NASA
CR
158

NASA CONTRACTOR REPORT

NASA CR-158

NASA CR-158

PRESTO

PROGRAM FOR RAPID EARTH-TO-SPACE TRAJECTORY OPTIMIZATION

by Robert E. Willwerth, Jr., Richard C. Rosenbaum, and Wong Chuck

Prepared under Contract No. NAS 1-2678 *by*
LOCKHEED MISSILES AND SPACE COMPANY
Sunnyvale, Calif.
for

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • FEBRUARY 1965

NASA CR-158

PRESTO

PROGRAM FOR RAPID EARTH-TO-SPACE TRAJECTORY OPTIMIZATION

By Robert E. Willwerth, Jr., Richard C. Rosenbaum, and Wong Chuck

Distribution of this report is provided in the interest of information exchange. Responsibility for the contents resides in the author or organization that prepared it.

Prepared under Contract No. NAS 1-2678 by
LOCKHEED MISSILES AND SPACE COMPANY
Sunnyvale, Calif.

for

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

For sale by the Office of Technical Services, Department of Commerce,
Washington, D.C. 20230 -- Price $5.00

TABLE OF CONTENTS

FOREWORD

This report was prepared by the Aerospace Sciences Laboratory of the Lockheed Missiles and Space Company, Sunnyvale, California. It presents the final documentation for the digital computer Program for Rapid Earth-to-Space Trajectory Optimization (PRESTO) developed by Lockheed for the Langley Research Center under NASA contract NAS1-2678. The computer program described herein was obtained from the NASA Request for Proposal L-2704, Computer Program for Powered Trajectory Optimization. The development of this program was administered by the National Aeronautics and Space Administration under the direction of Messers J. D. Bird and J. R. Elliott with Mr. A. E. Brown acting as contract monitor.

Development of the program presented in this report began in April 1963 and was completed in May 1964. Mr. R. E. Willwerth was responsible for program development. Optimization and control techniques were developed by R. C. Rosenbaum and the major portion of the programming was done by Wong Chuck. The work was performed under the cognizance of R. L. Nelson and D. I. Kepler of the LMSC Aerospace Sciences Laboratory.

This report together with a FORTRAN source listing binary object deck and symbolic deck concludes the work prescribed on contract NAS1-2678.

Abstract

The digital computer Program for Rapid Earth-to-Space Trajectory Optimization (PRESTO) uses a "closed loop" steepest descent optimization procedure to derive flight trajectories that produce maximum booster payloads for a variety of space missions. Trajectories can be computed in three degrees of freedom about a spherical rotating earth. Four powered stages and three upper stage thrust cycles are accommodated. Coast periods are permitted between each stage. Aerodynamic lift and drag forces are included in the computations.

Particular attention has been devoted during program development to computing speed. The convergence scheme, the general program arrangement and the subroutines have been selected with this in mind. In some cases standard subroutines have been modified to provide significant increases in speed for special applications. IBM 7094 computing times under one minute are currently being realized for complete three stage boost trajectory optimizations from the earth's surface to earth orbit.

The optimization routine simultaneously considers the launch direction and time, the interstage coast durations, the upper stage thrust sequencing, the complete pitch and yaw attitude histories and the terminal constraints. Intermediate constraints may be introduced on angle of attack, coast orbit perigee altitude or on the product of angle of attack and dynamic pressure. The closed loop procedure greatly facilitates the satisfaction of terminal constraints and reduces the number of iterations required to achieve convergence.

Four basic missions are accommodated by PRESTO; earth launch to earth orbit, earth launch to lunar transfer, lunar landing from lunar orbit and earth launch to interplanetary transfer. The lunar and interplanetary transfer missions can also be initiated from earth orbit. Terminal constraints for these missions have been developed in functional form. This allows the optimization of the combination of constraints required for specific missions along with the boost trajectory shape. From one to six terminal constraints are permitted on earth orbit and lunar landing missions, two or three constraints on lunar transfers and two constraints on interplanetary missions. Lunar, Mars, and Venus ephemeris routines are included in the program.

For orbit missions constraints are imposed either directly on the injection trajectory variables or on the orbit elements. When the orbit elements are specified, the functional constraint relationships are used. For lunar missions the constraint input is transfer time, day of launch and, if desired, transfer orbit inclination. The constraint routine defines the functional relationships between the injection trajectory variables that produce the required lunar intercept using special closed form expressions to represent the transfer orbit. For interplanetary missions the constraint inputs are transfer time and launch date. The departure asymptote direction and excess speed are computed from a matched conics routine and the functional relationships between the injection trajectory variables are defined in terms of the excess speed and the departure asymptote direction.

INTRODUCTION

The ability to evaluate the performance of multi-stage boost systems is a primary requirement for mission planning and preliminary design studies. The preliminary nature of the studies dictates a minimum expenditure of both time and money.

Unfortunately, large boost systems are usually limited by the propulsion units, and therefore deliver maximum payload with low stage thrust-to-weight ratios. This characteristic compounds the performance problem by increasing the sensitivity of performance to the "tilt" or thrust vector orientation program. Consequently the sensitivity of performance to changes in hardware parameters cannot be properly evaluated unless the effect of tilt program changes is eliminated by optimizing the tilt program for each set of parameters. In the past, optimization of tilt programs during a parametric preliminary design study has been almost prohibitively expensive.

Additional complications are introduced when the mission requires the satisfaction of a number of terminal or intermediate constraints. For some missions, these constraints may not be imposed directly on the trajectory variables but may only require that those variables be functionally related. Such is usually the case for interplanetary missions, for example, where the terminal constraints may be the hyperbolic excess speed and the departure asymptote direction. The position and velocity vectors at injection are then interrelated but not fixed, and the final combination of these vectors that is selected should be optimized along with the booster tilt program if maximum performance is desired.

Finally a number of parameters defining such quantities as the initial launch vectors, the coast periods between powered stages, and the upper stage thrust cycles may be treated as variables for certain boost hardware combinations. These "adjustable parameters" should then be optimized along with the tilt program and the terminal constraint combination to produce maximum performance.

Recently, procedures that allow rapid and economical evaluation of tilt histories have been developed. Recognizing the problem and the potential of the new techniques, NASA formulated a request (L-2704) for development of a digital program that will incorporate the new technology and provide a significant advance in capability. LMSC also recognized the potential of these developments and has vigorously pursued their exploitation. The new NASA program represents a natural extension of new optimization programs already developed by LMSC toward problems with additional degrees of freedom and more complicated constraints.

The digital computer Program for Rapid Earth-to-Space Trajectory Optimization (PRESTO) described in this document was developed to solve the performance problem outlined above for orbital, lunar and interplanetary missions. The formulation is such that solutions can be obtained in a single computer run with a minimum of computer time. Optimization of pitch and yaw tilt programs, constraint combinations and a variety of adjustable parameters is accomplished simultaneously with a closed loop steepest descent optimization routine.

A general description of the program organization is presented in Section 4 along with flow diagrams for the large subroutines. Symbols are defined in Section 5. Section 6 serves as a manual for operating the program. The data input and output formats are described and instructions are given for using the program options. The section concludes with a few comments on trouble shooting. All equations used in the program are given in Section 7. Section 8 provides a discussion of the theoretical methods used in the program and a detailed description of the programming of the basic optimization equations. Several of the subroutines in the program are of general interest and may be used in other programs. These are described in Section 9. Special program options are discussed in Section 10.

PROGRAM ORGANIZATION

In this section the overall format of PRESTO is presented. The sequence of trajectory iterations leading to an optimum is first described. Then the program organization, showing the general computation flow among the various subprograms, is discussed. Finally, lists of functions performed in each subroutine are provided along with block flow diagrams of some of the individual, more complicated, subroutines.

Sequence of Trajectory Iterations

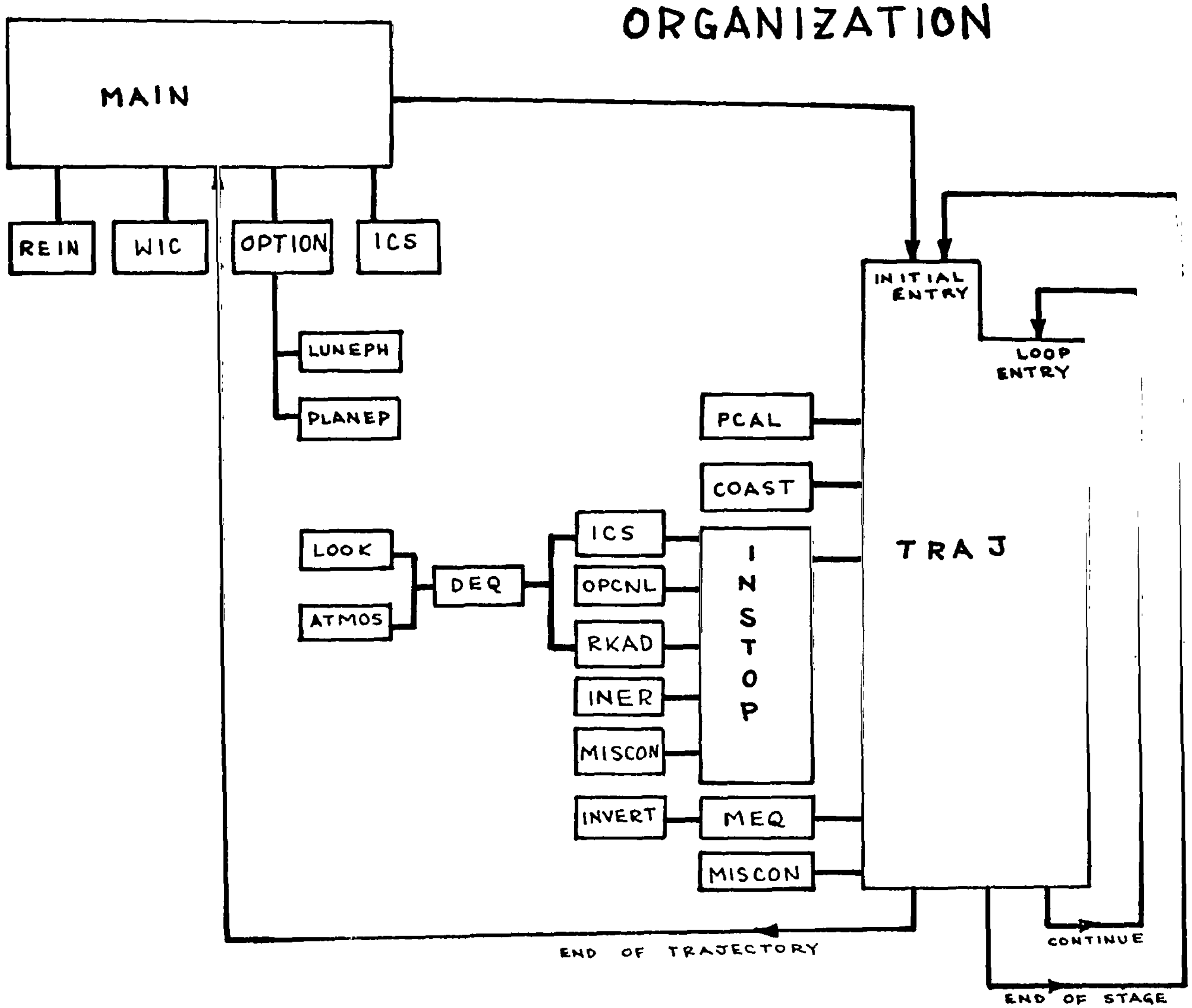
The several types of trajectory computations used in this program fall first into the categories of either "forward" or "backward." On a forward trajectory time proceeds positively and on a backward trajectory, negatively. On a forward trajectory only the equations of motion are integrated; on a backward trajectory the adjoint differential equations are solved. The next major categories are "guidance" or "optimization." On a guidance trajectory one is concerned only with meeting terminal constraints on the trajectory variables; in optimization, mass improvement is attempted as well.

Since solutions of the adjoint equations are used differently for forward guidance vs. optimization runs, a forward guidance run must be preceded by a backward guidance run. Similarly, a forward optimization run is preceded by a backward optimization run. Details of the differences between these are defined in Section 8 (Optimization Equations). The remaining major categories are "successful" or "unsuccessful," as judged at the end of each forward trajectory. A successful forward guidance trajectory satisfies terminal constraints within acceptable limits, and a successful forward optimization achieves some increase in terminal mass as well.

The sequence of trajectory iterations starts with the final category of run, the initial nominal. This, of necessity, is simply an "open-loop" integration of the equations of motion using an input thrust-direction history. This trajectory is then used as the basis for a backward guidance trajectory, which is followed by a forward guidance run. It is always assumed that a forward guidance trajectory represents an improvement over the previous nominal. Thus each forward guidance trajectory becomes the new nominal and the basis for a new solution of the adjoint equations. Backward and forward guidance runs are continued until one is judged "successful." At this point a backward optimization run is made and the magnitude of the initial mass improvement attempt is computed (see Section 8). A forward optimization run is made. If it is successful, it is used as the new nominal and backward and forward optimization trajectories are computed, etc. If it is unsuccessful, that trajectory is discarded and another forward optimization is computed with half of the previous attempt at mass improvement specified, etc. Thus, a succession of successful and unsuccessful optimization runs are computed either until the attempted mass improvement is smaller than a specified epsilon-weight or until the count of forward trajectories is within one of a specified limit. In either case, the sequence is then ended with a backward and a forward guidance trajectory.

PRESTO

PROGRAM SUBROUTINES AND GENERAL ORGANIZATION



P R E S T O

Sub-Program Names and Functions

1. MAIN Program

    Set constants
    Call REIN
    Call OPTION
    Compute integration intervals
    Set up flags for adjustable parameters
    Make lift-off calculation
    Set limits on differential equations to be integrated
    Determine what type of trajectory is to be computed
    Set up initial values for stage indices
    Compute initial delt-mass desired
    Increment launch and stage weights
    Call ICS
    Call TRAJ
    Check on meeting terminal constraints
    Update lunar position data
    Check on success of optimization
    Store characteristics of successful run

2. TRAJECTORY Subroutine

    Initial entry

    Set integration to Runge-Kutta
    Set initial stage and bring in stage data
    Set output frequency (vs. stage and type of trajectory)
    Pick up first point on backward trajectory

    Loop entry

    Test for time to set switch to compute change in adjustable parameter
    Adjust matrices row limits for storing and picking up:
        trajectory variables
        capital lambda's or B's or D's
    Call INSTOP
    Call PCAL
    Logic for coast perigee and circular park orbit constraints
    Call COAST
    Determine initial integration step size
    Compute P matrix
    Compute S matrix
    Test for end of stage and/or trajectory
    Call MEQ
    Return to loop entry or initial entry
    Call MISCON at end of trajectory

3. REIN Subroutine

    Read input data

4. OPTION Subroutine

Interrogate input options and set switches accordingly

Call LUNEPH or PLANEP

5. LUNEPH (Lunar Ephemeris) Subroutine

Compute lunar position for specified arrival date

6. PLANEP Subroutine

Compute planetary positions for specified trip
Compute required departure hyperbolic excess velocity vector

7. MISCON (Mission Constraints) Subroutine

Compute current value of stopping parameter
Compute achieved values of terminal constraints

8. ICS Subroutine

Assign initial conditions to trajectory and adjoint variables

Call DEQ

9. PCAL (Preliminary Calculations) Subroutine

Compute perturbations in control variables
Apply control variable constraints
Compute chantes in adjustable parameters
Check for switch between closed-loop and open-loop
Compute vector EK

10. ATMOS Subroutine

Compute atmospheric density, pressure and speed of sound
Compute partial derivatives re: atmosphere for adjoint equations

11. INSTOP Subroutine

Call ICS
Call INER
Store first point when going forward
Call OPCNL
Call RKAD
Pick up stored trajectory variables when integrating backward
Stop integration when stopping parameter is reached
Call MISCON

12. RKAD Subroutine

Logic for integration using both Runge-Kutta and Adams integration

13. DEQ Subroutine

Calculate thrust and aerodynamic forces using ATMOS and LOOK
Check whether trajectory equations and/or adjoint equations are required
Compute time derivatives

14. MEQ (Matrix Equations) Subroutine

Evaluate trajectory deviations
Store trajectory variables
Generate lambda and I matrices
Compute A matrix
Invert A matrix (Call Subroutine INVERT)
Compute B, D and C, E matrices
Compute payload change for fixed final stage option

15. INVERT Subroutine

Invert Matrix

16. LOOK Subroutine

Linear interpolation table look up

17. OPCNL (output control) Subroutine

Write output

18. COAST Subroutine

Compute trajectory variables in closed form
Compute discontinuities in adjoint variables

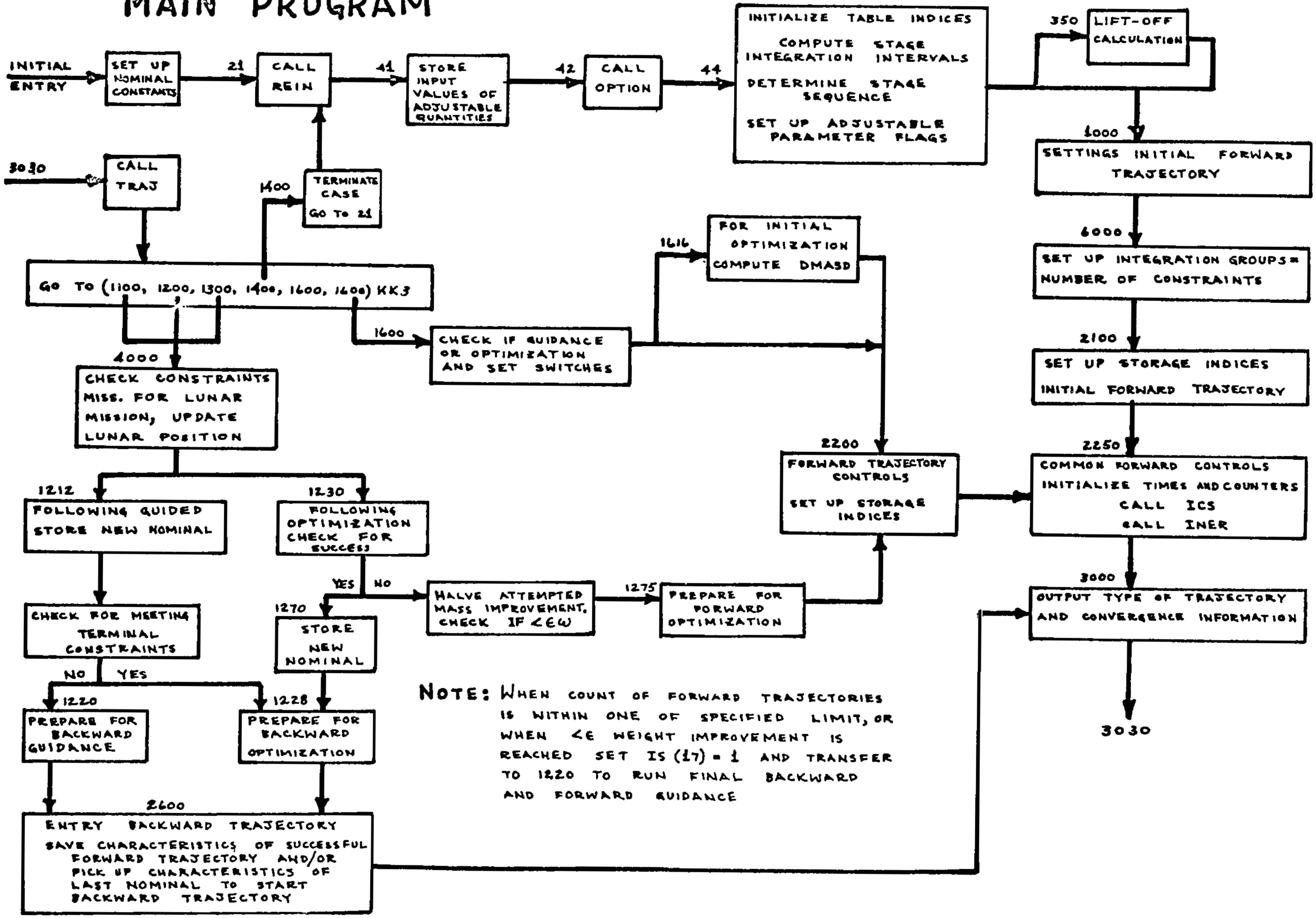
19. INER Subroutine

Compute inertial trajectory variables

20. WIC Subroutine

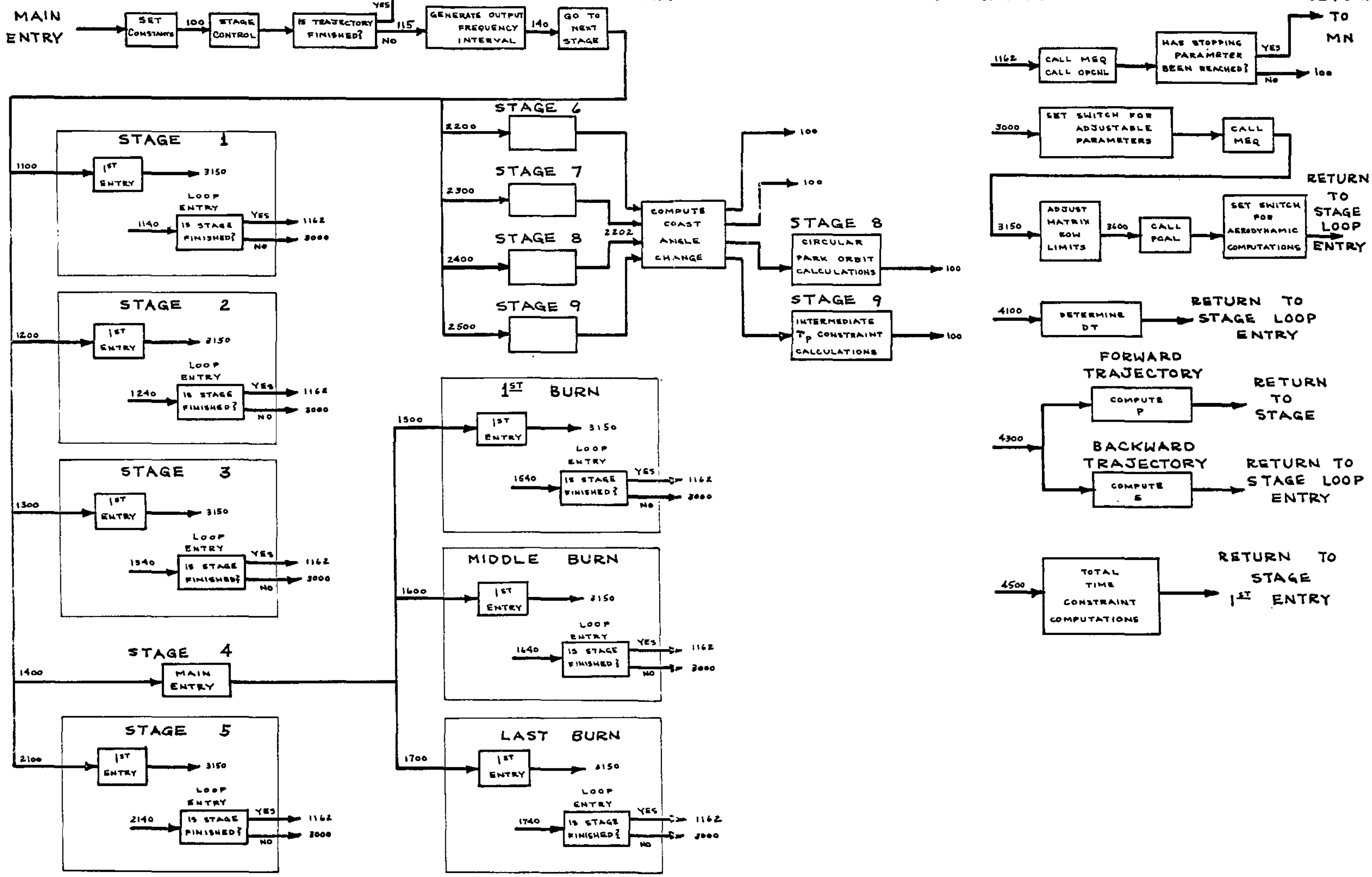
Write out input data

MAIN PROGRAM

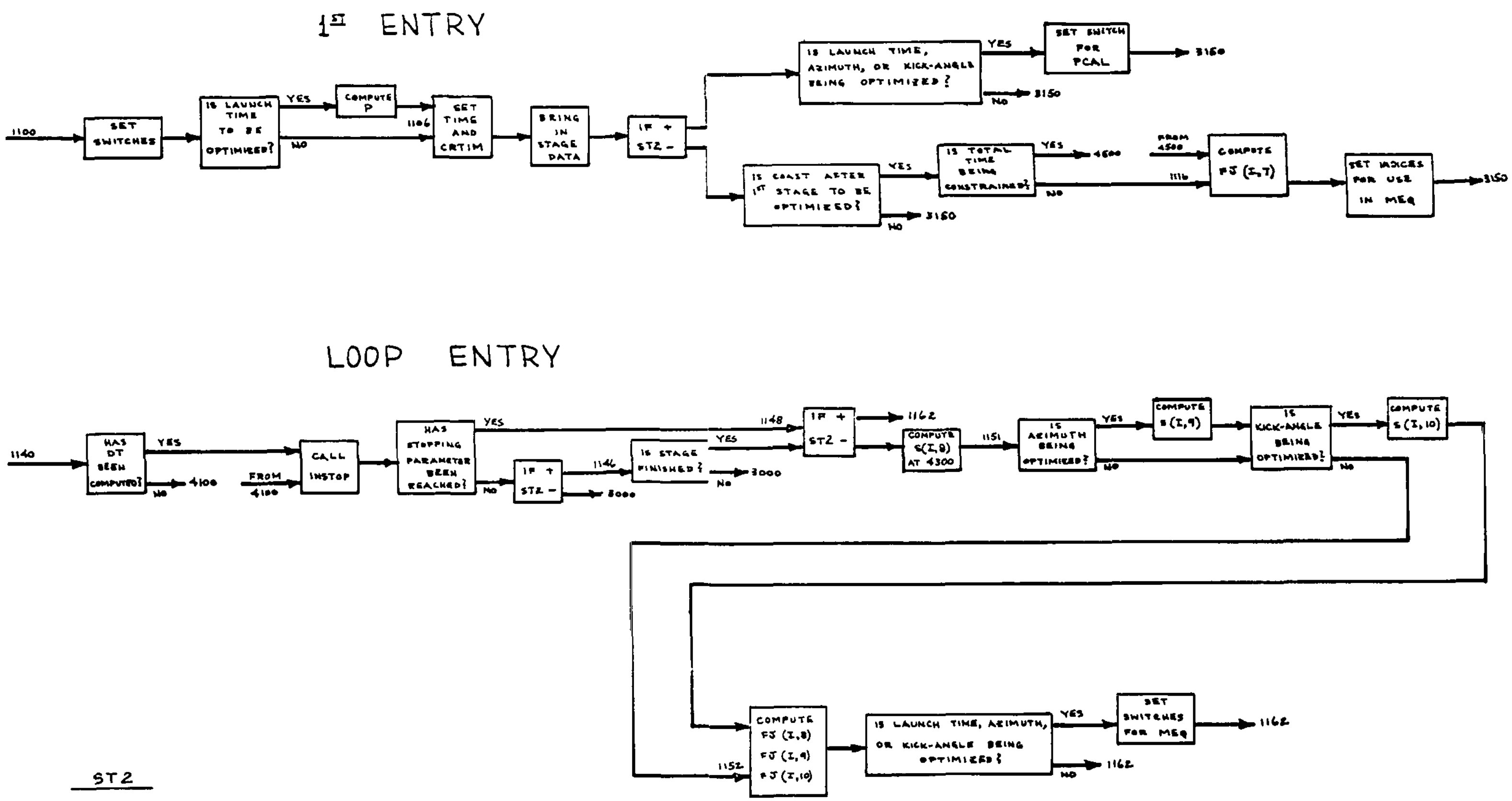


4-7

TRAJECTORY FLOW DIAGRAM



TRAJECTORY- STAGE ONE FLOW DIAGRAM

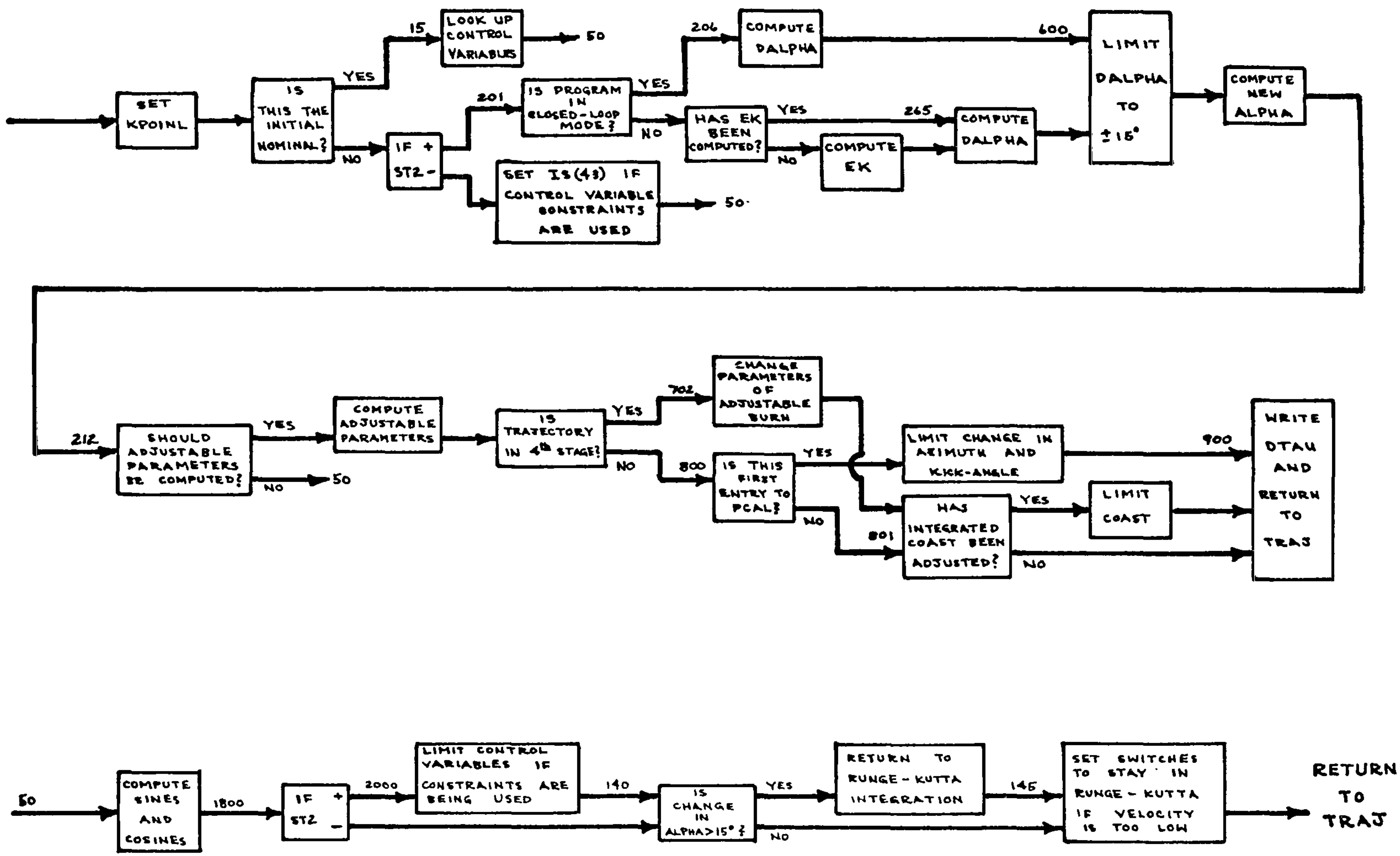


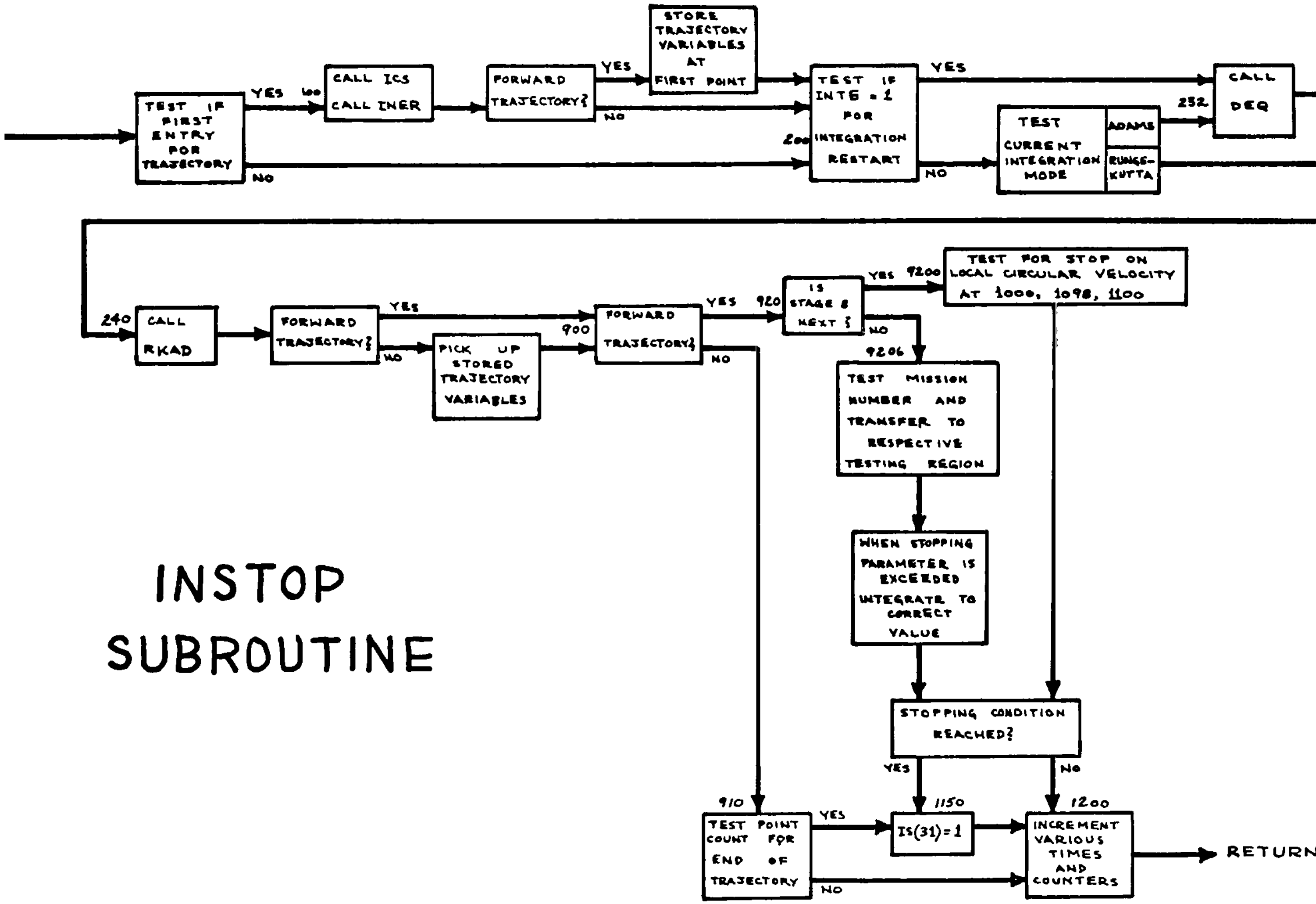
ST2

+ FORWARD TRAJECTORY

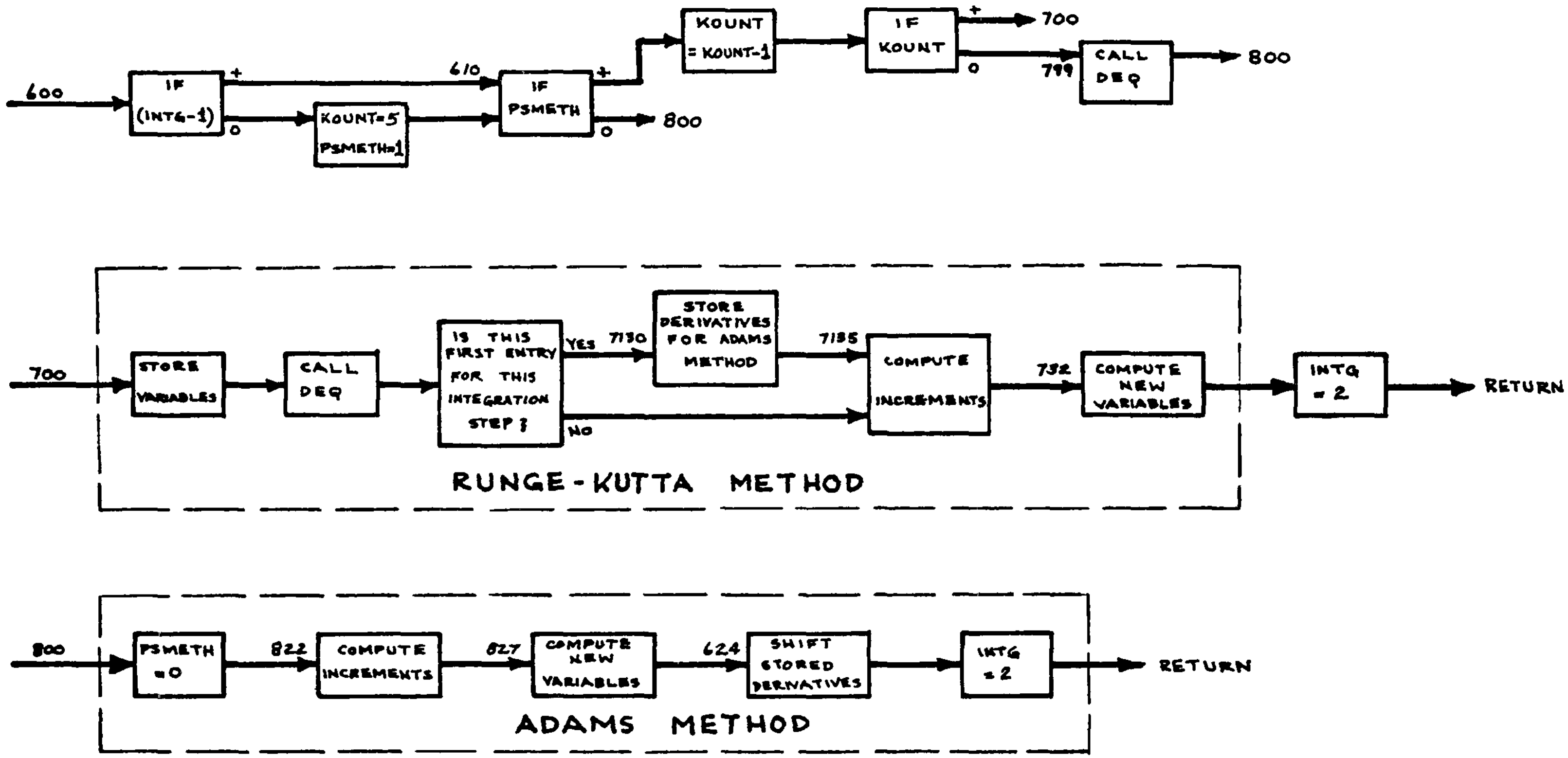
- BACKWARD TRAJECTORY

PCAL SUBROUTINE





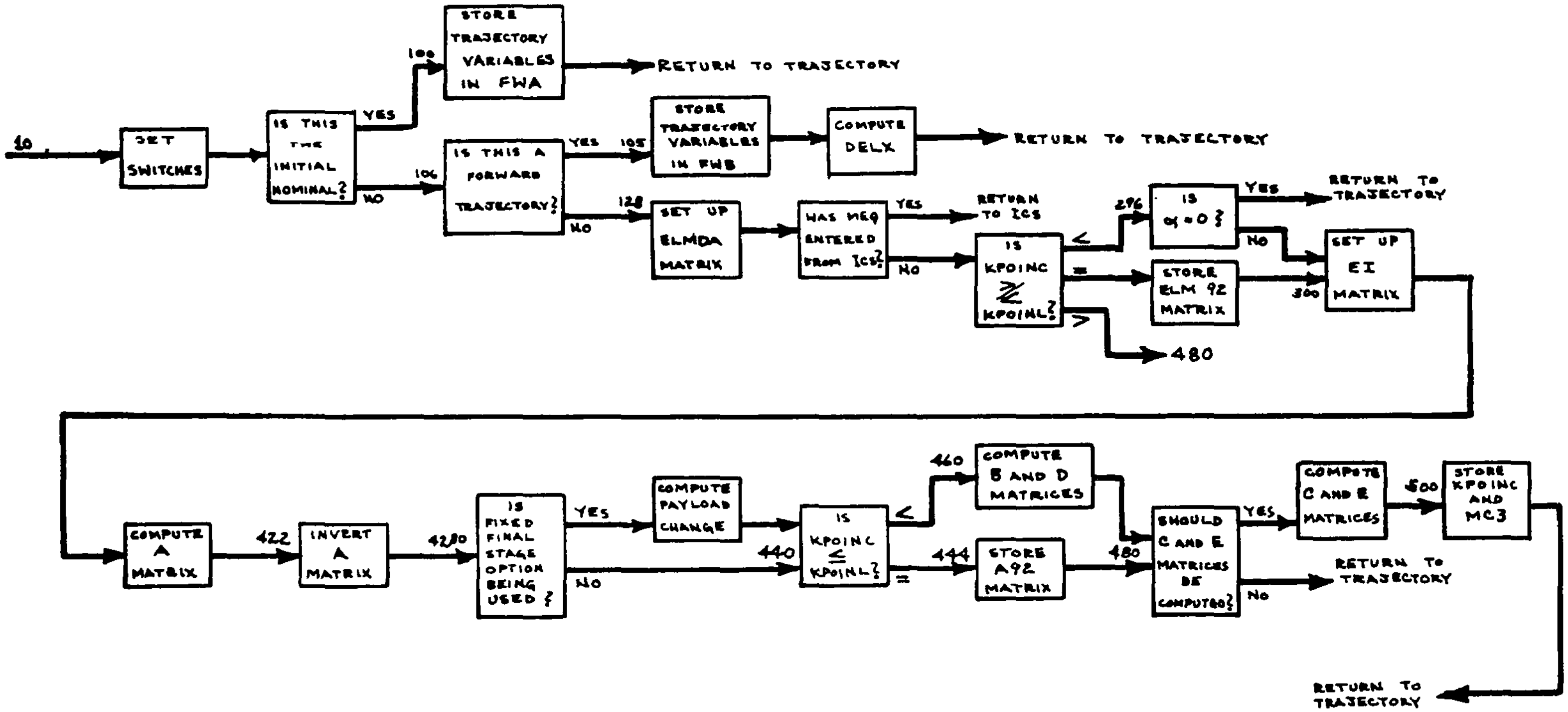
INSTOP SUBROUTINE



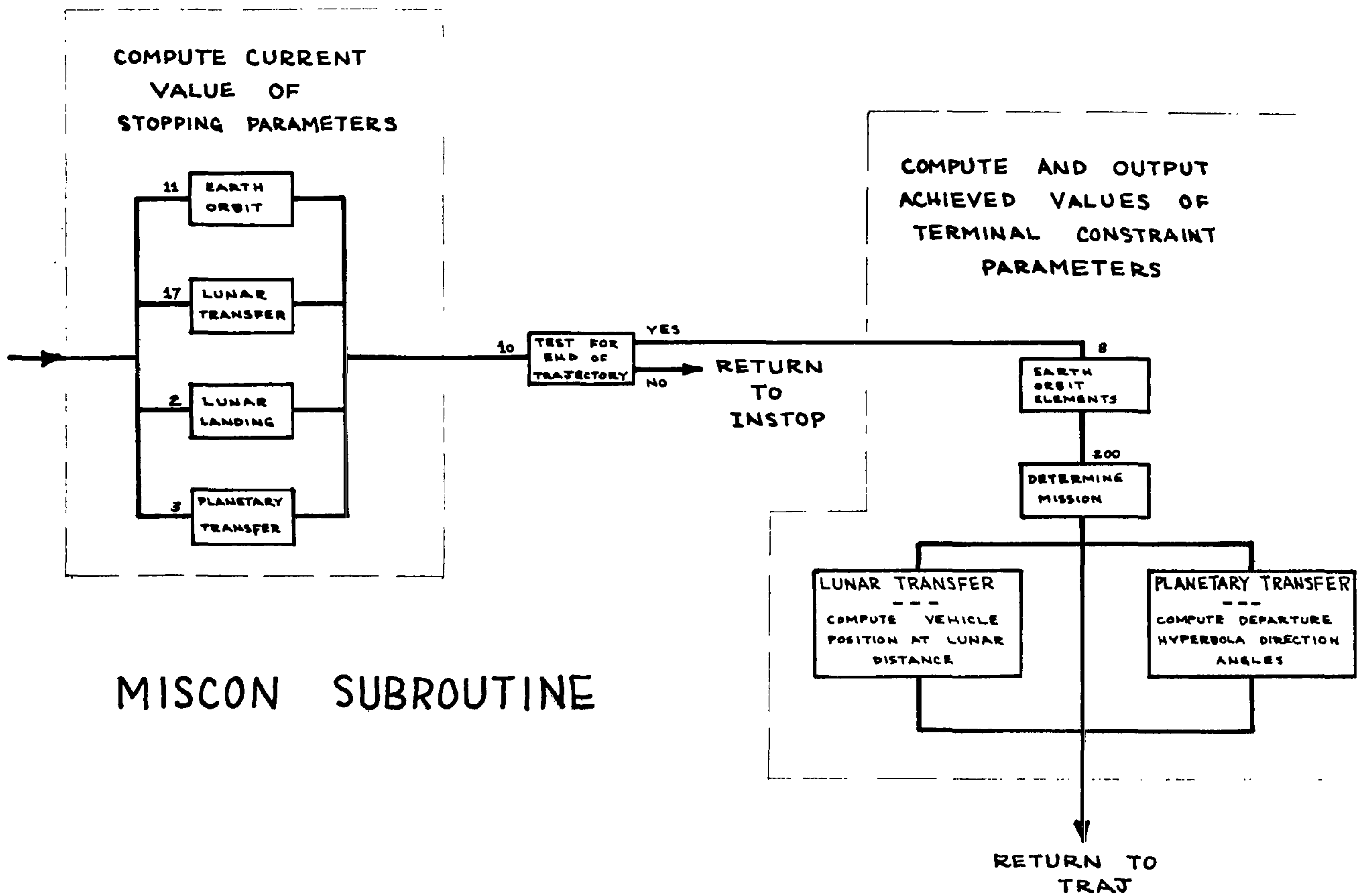
NOTE:

1. INTG = 1 WHEN RUNGE-KUTTA INTEGRATION BEGINS.
2. EXCEPT FOR THE FIRST POINT OF INTEGRATION, THE DERIVATIVES REQUIRED FOR THE ADAMS METHOD ARE CALLED FOR FROM INSTOP.

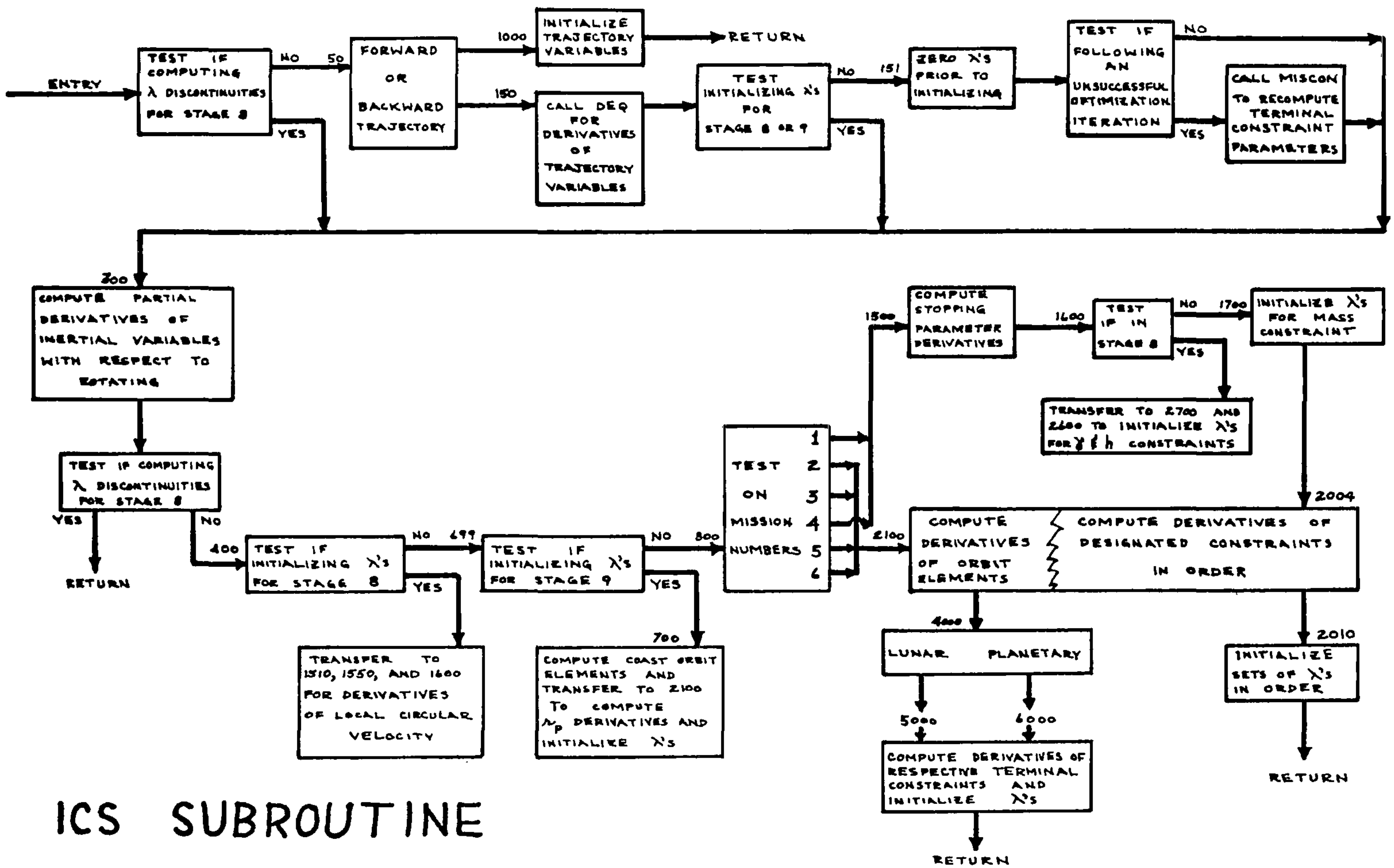
RKAD SUBROUTINE



MEQ SUBROUTINE



MISCON SUBROUTINE



ICS SUBROUTINE

CODING NOMENCLATURE

AA	A matrix
A92	A matrix at start of open loop computation
ACTN	answer from 4-quadrant arctan routine
AEH	right ascension of VH vector
AEM	right ascension of vehicle position @ lunar radial distance
ALIT	local speed of sound
ALPHAE	right ascension of present position
ALPHA	net angle of attack from ETA and CHI
B	B matrix
BEH	BETAH--see planetary mission terminal constraints equations
BEM	BETAM--see lunar mission terminal constraints equations
BETA	in-plane range angle from ascending node
BETA 2	incremented BETA in closed-form coast equations
BETA C	range angle in closed-form coast stages
BETAP	argument of perigee
BFA 6	first buffer for tape 1
BFB 6	second buffer for tape 1
CCHI	cosine of CHI
CCHIQ	$(CCHI)^2$
CD	total drag coefficient
CEL 2	in the calling sequence for buffer output
CEL 6	in the calling sequence for buffer output
CETA	cosine of ETA

CETAQ	$(\text{CETA})^2$
CGAM	cosine of GAMMA (flight path angle)
CGAMI	cosine of inertial flight path angle
CHI	thrust yaw angle
CHIRT	function of CHI and ETA in PCAL
CLAM	cosine of latitude
CL	lift coefficient
CPHI	cosine of PHI (roll angle)
CPSI	cosine of PSI (azimuth)
CPSII	cosine of inertial azimuth
CPSIIQ	$(\text{CPSII})^2$
C	C matrix
CRTIM	time for end of stage
CRTIM1	time for end of 1st burn in stage 4
CRTIM2	time for end of 2nd burn in stage 4
CT1	Data block 10
CT2	Data block 11
CT3	Data block 12
CT4	Data block 13
CT5	Data block 14
CT6	Data block 15
CTAU	cosine of longitude
D	drag
D1	first set of derivatives used in integration
D2	second " " " " " "
D3	third " " " " " "
D4	fourth " " " " " "

DA0	Available data block (18)
DA1	Data block 16
DA2	" " 17
DA4	" " 19
DA5	" " 20
DA6	" " 21
DA7	" " 22
DA8	" " 23
DA9	" " 24
DA10	" " 25
DA11	Available data block (26)
DA12	Data block 27
DA13	" " 28
DA14	" " 29
DAH	derivative of ALIT/altitude
DALPHA	change to be made in control variable (s) ALPHA
DATE	date
DBETAC	adjustment in BETAC
DD	D matrix
DEH	declination of VH vector
DELTAY	increments in X's from integration routine
DELX	deviation(s) from nominal trajectory
DEM	declination of vehicle position @ lunar radial distance
DIF	used in numerical formulation of partial derivatives across closed-form coast
DPDWFUEL	partial derivative relating payload to fuel remaining in final stage
DPDPSI(I)	partial derivative relating payload to error in terminal constraint (I)

DMASD	calculated initial mass improvement
DP2	Estimate of integral of square of control deviations
DPH	derivative of pressure/altitude
DPSI	negative of error in terminal constraints on latest nominal trajectory
DPSIS	negative of error in terminal constraints on latest trajectory
DRH	derivative of density/altitude
DTAU	change to be made in adjustable parameter(s)
DTOLD	integration interval from previous step
DT	current integration interval
DWFUEL	propellant remaining at trajectory termination
DWP	increment to be added to payload
DWPMAX	maximum change in payload at any one time
E	E matrix
EH	angular momentum
EI	I matrix
EK	K vector
ELAM2	Latitude at end of coast
ELM92	λ matrix at start of open loop computation
ELMDA	λ matrix
EML	mass at end of closed-form lift-off
ENU	angle NU, longitude position from node
ENU2	angle NU at end of coast
ENUH	angle NU to VH vector
ENUM	angle NU to lunar outward radial
ETA	ALPHA component in vertical plane
EYE	orbit inclination

FJ	J matrix
FL	lift
FMU	μ, gravity constant
FMUDEH	μ/EH
FWA	storage area of latest nominal trajectory
FWB	storage area of latest trajectory
G	local acceleration due to gravity
GAMI	inertial flight path angle
GAMI2	inertial flight path angle at end of coast
GO	weight-to-mass conversion factor
H	altitude
HAERO	altitude above which aerodynamic computations are bypassed
HCAM	MACH number
HEAD1	data block 1
HEAD 2	data block 2
HL	altitude at end of closed-form lift-off
HSTR	geopotential altitude
I	used as index locally
IBO	data block 7 (available)
IB1	" " 6
IB3	" " 8
IB4	" " 9
IBL	Index used in read-in subroutine
IC	1 for optimization; 2 for guidance

ICT(n) SUBSCRIPTED INTEGER SIGNAL FLAGS FOR TERMINAL CONSTRAINTS

ICT (1)	...	0	=	One	Constraint	
		1	=	No	..	
ICT (2)	...	0	=		Two	Constraints
		1	=	Less than	Two	..
ICT (3)	...	0	=		Three	..
		1	=	Less than	..	..
ICT (4)	...	0	=		Four	..
		1	=	Less than	..	..
ICT (5)	...	0	=		Five	..
		1	=	Less than	..	..
ICT (6)	...	0	=		Six	..
		1	=	Less than	..	..

INCODE	flag for buffered output routine
IO1	field indicators for data header cards
IO2	" " " " " "
IO3	" " " " " "
IO4	" " " " " "
IP	data block 4; see list of options
IS21	0 = no circular park orbit 1 = circular park orbit (coast stage 8) is being used

RKAD INTEGRATION PACKAGE ENTRY FLAGS

INTG ... 1 = To start or re-start integration

A. Method of Integration .. RUNGE-KUTTA

B. Initial entry .. set by user

C. Subsequent entries .. set by RKAD

2 = Normal continuation of integration

A. Set by RKAD

IS(n) SUBSCRIPTED INTEGER SIGNAL FLAGS

IS (1)	...	0 =	2 Degrees of freedom
		1 =	3 Degrees of freedom
IS (2)	...	0 =	Rotating planet
		1 =	Non-rotating planet
IS (3)	...	0 =	Zero density (Drag = Lift = 0)
		1 =	Non-zero density
IS (4)	...	0 =	Zero drag
		1 =	Non-zero drag
IS (5)	...	0 =	Zero lift
		1 =	Non-zero lift
IS (6)	...	0 =	To by-pass option selections
		1 =	To process option code
IS (7)	...	0 =	Correct flag
		1 =	Wrong flag
IS (8)	...	0 =	Correct sub-options
		1 =	Error in option code
IS (9)		- 1 =	Error exit from REIN
		0 =	Proceed to case computations
		1 =	End of job
IS (10)	...	1 =	To comput adjoint equations
		0 =	To by-pass
IS (11)	...	- 1 =	Use past partial DT
		0 =	Compute and use stage DT
		1 =	Use stage DT

IS (12)	...	Used in several places in connection with circular park orbit option
IS (13)	...	0 = Do not use coast equations for circular orbit 1 = Use coast equations for circular orbit
IS (14)	...	-1 = Altitude constraint on circular park orbit has just been violated 0 = Altitude constraint on circular park orbit has not been violated 1 = Altitude of circular park orbit is being constrained
IS (15)	...	0 = Unsuccessful forward trajectory 1 = Successful forward trajectory
IS (16)	...	0 = DMASD has not been computed yet 1 = DMASD has been computed
IS (17)	...	0 = More forward runs are allowed 1 = No more forward runs allowed
IS (18)	...	0 = Program is in closed-loop mode 1 = Program is in open-loop mode
IS (19)	...	(AV)
IS (20)	...	Upper limit for page line count
IS (21)	...	Temporary value for KPOINL
IS (22)	...	Plus = T9 point count is greater than or equal to input value (after storage). Zero = T9 point count is less than input value. Negative = T9 point count is greater than or equal to input value.

IS (23)	...	Current output frequency
IS (24)	...	Current output count
IS (25)	...	0 = By-pass PF stage 4 critical time computations
		1 = Compute PF stage 4 critical times
IS (26)	...	Storage for IS (10)
IS (27)	...	0 = Normal outputs
		1 = By-pass initial outputs on initial stage
IS (28)	...	0 = By-pass zero aerodynamic computations
		1 = Compute zero aerodynamic terms
IS (29)	...	0 = Search for time to start C and E matrices comp.
		1 = By-pass time search
IS (30)	...	Current successful forward trajectories count
IS (31)	...	0 = Stopping parameter test
		1 = Reached stopping condition
IS (32)	...	0 = Failed to meet terminal conditions (Guidance trajectory)
		1 = Terminal conditions have been met (Guidance trajectory)
IS (33)	...	-1 = Call ICS from MAIN
		0 = Call ICS from INSTOP
		1 = Do not call ICS
IS (34)	...	Storage (used in ICS)
IS (35)	...	0 = Compute inertial quantities only
		1 = Compute interial quantities and their derivatives

IS (36)	...	= IS (5)
IS (37)	...	0 = Buffered output
		1 = Non-buffered output
IS (38)	...	0 = Call MISCON -- Compute only stopping parameter
		1 = ... -- Compute all constraints
IS (39)	...	4th PF burns
IS (40)	...	(AV)
IS (41)	...	-1 = r_p constraint has been violated on current forward run
		0 = r_p constraint has not been violated
		+1 = r_p is being constrained to input value
IS (42)	...	0 = Normal entry to ICS
		1 = Entry to ICS to set up adjoint variable for r_p constraint
IS (43)	...	0 = No α constraint over this time interval
		1 = α constraint applied; zero D1(I), I = 44,64
IS (44)	...	0 = Stage time less than critical time on last successful run
		1 = Stage time greater than critical time on last successful run
IS (45)	...	Change in number of integration steps during first burn of 4th stage as compared to last successful run
IS (46)	...	Change in number of integration steps during second burn of 4th stage as compared to last successful run
IS (47)	...	0 = S matrix has not been adjusted if total time constraint is being used
		1 = S matrix has been adjusted
IS (47)	...	0 = MEQ is called from an intermediate point in a stage
		1 = MEQ is called from the final point in a stage
IS (48)	...	0 = Call MEQ from region 1 of TRAJ
		1 = Don't call MEQ
IS (49)	...	0 = KPOINL has not yet been computed
		1 = KPOINL has been computed
IS (50)	...	-1 = An adjustable coast has been eliminated because it went to 0
		0 = An adjustable coast has not been eliminated
		1 = An adjustable coast has gone to 0

ITAP(1)	0 = to output tape 2
	1 = by-pass tape 2 outputs
ISTGE	data block 5
IT1	non-subscripted flag
IT2	"
IT3	"
IT4	"

**NON-SUBSCRIPTED INTEGER (TABLE INDICES)

JTT1	...	Current Linear-Segment Index for Table TT1
JTT2	...	.. .. TT2
JTT3	...	.. .. TT3
JTT4	...	.. .. TT4
JTD1		TD1
JTD2		TD2
JTD3		TD3
JTD5		TD5
JTL1		TL1
JTL2		TL2
JTL3		TL3
JTL5		TL5
JTK1		TK1
JTK2		TK2
JTK3		TK3
JTK5		TK5
JTTH		TTH
JTET		TET
JTCH		TCH
JTMD		TMD
JTMS		TMS
JMOS		Blocks CT2, CT3, CT4 and CT5

J	used as index locally
JC	number of constraints
JC1	" " "
K	used as index locally
KCEPOI	integration step at which adjustable parameters are computed
KFLAG	flag indicating use of adjustable parameter(s)

KKn NON-SUBSCRIPTED INTEGER SIGNAL FLAGS

KK1	...	Current stage index			
KK2	...	Current stage code			
		1	=	Powered flight 1.	
		2	=	.. .. 2.	
		3	=	.. .. 3.	
		4	=	.. .. 4.	
		5	=	Coast 1.	
		6	=	.. 2.	
		7	=	.. 3.	
		8	=	.. 4.	
		9	=	.. 5.	
KK3	...	1	=	Initial	Forward trajectory
		2	=	Intermediate	.. ..
		3	=	Intermediate	.. ..
		4	=	Last	.. ..
		5	=	(AV)	
		6	=	Backward trajectory	

KK4	...	Current stage code
		1 = Powered flight stage 1.
		2 = ... 2.
		3 = ... 3.
		4 = First burn (PF stage 4).
		5 = Middle burn (PF stage 4).
		6 = Last burn (PF stage 4).
		7 = Coast stage 1.
		8 = ... 2.
		9 = ... 3.
		10 = ... 4.
		11 = ... 5.
KK5	...	-1 = Backward trajectory
		+1 = Forward trajectory
KK6	...	Number of stages in the trajectory
KK7	...	Forward trajectory ... KK7 = KK6
		Backward trajectory... KK7 = 1
KK8	...	Current index for selecting aerodynamic constants
KK9	...	Storage index computed during initial forward run
KK10	...	Number of 4th powered flight burns
KK11	...	3 = first Forward trajectory
		2 = Intermediate .. ..
		1 = Last .. ..
KK12	...	(AV)
KK13	...	Non-subscripted integer form of IP (3) + 1.

KK14	...	Current 4th powered flight burn
KK15	...	Increment of KK14
KK16	...	- 1 = Initial forward trajectory 0 = Optimization run 1 = Guidance run
KK17	...	Return code
KK18	...	Non-scripted integer form of IP (16) +1
KK19	...	Return code for assign go to statements
KK20	...	Last stage
KK21	...	A. Forward trajectory ... KK21 = past stage B. Backward trajectory... KK21 = next stage
KK22	...	Storage for current K (K indicates the current Kth adjustable parameter)
KK23	...	Number of remaining 4th powered flight burns of the last successful forward trajectory
KK24	...	Index for DTAU selection
KK25	...	KK25 = 5 - KK3

KK26 ... last coast stage

KK27 ... past stage on backward trajectory

KK28 ... adjustable parameter to be eliminated

KK29 ... KK4-7 at time of elimination of adjustable coast because it went to zero

KK30 ... 0 Evaluate derivatives in DEQ

1 Evaluate forces only in DEQ

$\left.\begin{array}{l}\text{KK31}\\\text{KK32}\end{array}\right\}$ ... Temporary use for r_p constraint with adjustable burn

KKONE	value of KK1 at beginning of backward trajectory
KPOINC	count of integrating points from beginning of trajectory
KPOINL	integration point count to switch from closed to open loop
KPOINS	number of integration points on last successful forward run
L	used as index locally
L1	"
L2	"
L3	"
L4	"
LAGE	
LBD	limit on storage index for B and D matrices
LCE	" " " " " C and E "
LDEQ	" " number of integration variables
LFWAB	" " storage in FWA and FWAB
LINE	current page output line count
M	used as index locally
MB1	upper index for storing in B matrix region
MB2	lower index for storing in B matrix region
MB3	increment in MB1 and MB2 per integration step
MC1	upper index for storing in C matrix region
MC2	lower indexfor storing in C matrix region
MC3	increment in MC1 and MC2 per storage point
MC4	number of points at which C and E Matrices are stored
MF1	upper index for storing or picking up in FWB

MF2	lower index for storing or picking up in FWB
MF3	increment in MF1 and MF2 per integration step
ML1	upper index for storing Λ matrix
ML2	lower index for storing Λ matrix
ML3	increment in ML1 and ML2
N	used as index locally
NB1	upper index for picking up from B matrix region
NB2	lower index for picking up from B matrix region
NB3	increment in NB1 and NB2 per integration step
NC1	upper index for picking up from C matrix region
NC2	lower index for picking up from C matrix region
NC3	increment in NC1 and NC2 per storage point
NC3S1	(-NC3) for particular stage
NC4	number of points at which adjustable parameters are computed
NF1	upper index for storing or picking up in FWA
NF2	lower index for storing or picking up in FWA
NF3	increment in NF1 and NF2 per integration step
NL1	upper limit for picking up Λ matrix
NL2	lower limit for picking up Λ matrix
NL3	increment in NL1 and NL2
NT	tape number
OMEGA 2	Earth rotation rate
OMEGA	2 * OMEGA
OMEGAE	longitude of ascending node
OMEGAQ	$(\text{OMEGA})^2$

P(I,J)	matrix of derivatives of trajectory variables
P	local atmospheric pressure
PARDT(I)	size of final integration step for parts of 4th stage
PARDTI	stored values of PARDT(I)
PDA	partial derivative of drag/ALPHA
PDM	partial derivative of drag/MACH
PF	partial derivatives of DV/dt
PG	" " " $d\gamma/dt$
PH	" " " $d\psi/dt$
PI	" " " dr/dt
PJ	" " " $d\lambda/dt$
PK	" " " $d\tau/dt$
PLA	partial derivative of lift/ALPHA
PLM	" " " lift/MACH
PO	sea level atmospheric pressure
PSII	azimuth of VI
PSII2	azimuth of VI at end of coast
PS(I,J)	value of P(I,J) on last successful forward run
PSX(I)	$\partial S/\partial X$ (for circular park orbit)
PSMETH	flag in RKAD indicating (past) method integration for previous point
QBAR	dynamic pressure
QBARA	ALPHA * QBAR
QCH, QD, QET, QK, QL, QT, QTH	quotients (slopes of linear segments) in tables using corresponding J's

R2	radius at end of coast
RALPHA	orbit semi-major axis
RE	radius of earth
RHO	local atmospheric density
RM	radial distance to moon
RP	orbit perigee radius
RSTR	gas constant for air
S	S matrix
SCHI	sine of CHI
SCHIQ	$(SCHI)^2$
SDA1	
SETA	sine of ETA
SETAQ	$(SETA)^2$
SG1	stage 1 data (data block 39)
SG2	" 2 " "
SG3	" 3 " "
SG4	" 4 " "
SG5	" 5 " "
SGAM	sine of γ
SGAMI	sine of GAMI
SGNET	sign of ETA
SLAM	sine of latitude
SPHI	sine of PHI (roll angle)
SPSI	sine of PSI (azimuth)
SPSII	sine of PSII

ST1(n) CURRENT-STAGE DATA

ST1 (1)	...	FISP	...	Specific impulse (Vac.)
ST1 (2)	...	A	...	Aerodynamic reference area
ST1 (3)	...	AE	...	Nozzle exit area
ST1 (4)	...	MO	...	Initial weight
ST1 (5)	...	MF	...	Final weight

ST2(n) PERMANENT STORAGE

ST2 (1)	...	ST2 (1) = +1 For forward trajectory ST2 (1) = -1 For backward trajectory
ST2 (2)	...	To count the number of forward trajectories
ST2 (3)	...	Storage (used in instop)
ST2 (4)	...	Storage (used in instop)
ST2 (5)	...	Storage (past DT)
ST2 (6)	...	Storage (current DT)
ST2 (7)	...	Storage (past time)
ST2 (8)	...	(AV)
ST2 (9) : ST2 (15)	...	Lunar position data used in LUNEPH and MAIN .. ..
ST2 (16)	...	DWFUEL multiplicative factor
ST2 (17)	...	TIMEC (to transfer between subroutines)
ST2 (18)	...	(AV)
ST2 (19)	...	Final TIMCT - desired value (for total time constraint)
ST2 (20)	...	TIMCT - TIMT at beginning of last coast on last successful forward run (for total time constraint)
ST2 (21)	...	TIMCT - TIMT at beginning of last coast on current run (for total time constraint)
ST2 (22)	...	DWPMAX
ST2 (23)	...	DA12(11)
ST2 (24)	...	DA12(2)
ST2 (25)	...	DA12(4)

ST3(n) STORAGE FROM TERMINAL CONSTRAINT COMPUTATIONS

ST3 (1)	...	Achieved values of terminal constraint quantities
.		
.		
.		
ST3 (6)	...	.. ..
ST3 (7)	...	(Available)
.		
.		
.		
ST3 (10)	...	(Available)

ST4(n) STORAGE OF INPUT DATA

ST4 (1)	...	= DA9 (1)
ST4 (2)	...	= DA9 (2)
ST4 (3)	...	= DA9 (3)
ST4 (4)	...	= DA9 (4)
ST4 (5)	...	= DA1 (9)
ST4 (6)	...	= DA1 (10)
ST4 (7)	...	= DA1 (11)
ST4 (8)	...	= SG1 (4)
ST4 (9)	...	= SG1 (5)
ST4 (10)	...	= SG2 (4)

ST4 (11)	...	=	SG2 (5)
ST4 (12)	...	=	SG3 (4)
ST4 (13)	...	=	SG3 (5)
ST4 (14)	...	=	SG4 (4)
ST4 (15)	...	=	SG4 (5)
ST4 (16)	...	=	SG5 (4)
ST4 (17)	...	=	SG5 (5)
ST4 (18)	...	=	P(1,1)
ST4 (19)	...	(AV)	
ST4 (20)	...	(AV)	

ST5(n) STORAGE FOR ADJUSTABLE PARAMETERS

ST5 (1)	...	DTAU Storage
.		
.		
.		
ST5 (10)	...	.. ..

STAU	sine of longitude
STGH	stage integration intervals
SX	saved values of X(n)
T	thrust
TANAG	tangent of the angle
TANN	numerator of TANAG
TAU2	longitude at end of coast
TCH	tables identified by J's
TD	
TET	
TK	
TLN	
TT	
TTH	
TG	current time in space age date
TGO	launch time " " " "
TGM	lunar arrival time in space age data
THETA	thrust angle to horizontal
TIMCT	time from launch including coasts
TIMCTS	saved value of TIMCT
TIME	time from beginning of present stage exclusive of coasts
TIMEC	time in integrated coast
TIMEP	CT1(10)
TIMT	time from launch excluding coasts
TIMTS	saved value of TIMT
TL	time duration of closed-form lift-off
TM	transfer time to the moon (hours)
TO	initial thrust for lift-off calculation

TVL	thrust at end of lift-off calculation
TV	vacuum thrust
TVO	initial vacuum thrust
TWOE	twice the total energy of orbit
TZETA	time from perigee at beginning of coast
TZETA2	time from perigee at end of coast
VH	hyperbolic excess velocity
VI	inertial velocity
VI2	inertial velocity at end of coast
VL	velocity at end of closed form lift-off
W	weight as measured at sea level
X	variable of integration as follows:

VARIABLES OF INTEGRATION

X (1)	=	V	(Velocity)
X (2)	=	GAM	(Flight path angle)
X (3)	=	R	(Radius)
X (4)	=	M	(Mass)
X (5)	=	TAU	(Longitude)
X (6)	=	PSI	(Azimuth)
X (7)	=	LAM	(Latitude)
X (8)	=	LV1	(Adjoint Variables)
X (9)	=	LG1	.
X (10)	=	LR1	.
X (11)	=	LM1	.
X (12)	=	LV2	
X (13)	=	LG2	
X (14)	=	LR2	
X (15)	=	LM2	
X (16)	=	LV3	
X (17)	=	LG3	
X (18)	=	LR3	
X (19)	=	LM3	
X (20)	=	LV4	
X (21)	=	LG4	
X (22)	=	LR4	
X (23)	=	LM4	
X (24)	=	LP1	
X (25)	=	LL1	

X (26)	=	LP2	(Adjoint variables)
X (27)	=	LL2	.
X (28)	=	LP3	.
X (29)	=	LL3	.
X (30)	=	LP4	
X (31)	=	LL4	
X (32)	=	LV5	
X (33)	=	LG5	
X (34)	=	LR5	
X (35)	=	LM5	
X (36)	=	LP5	
X (37)	=	LL5	
X (38)	=	LV6	
X (39)	=	LG6	
X (40)	=	LR6	
X (41)	=	LM6	
X (42)	=	LP6	
X (43)	=	LL6	
X (44)	=	EI11	(Capital I's)
X (45)	=	EI12	
X (46)	=	EI22	
X (47)	=	EI13	
X (48)	=	EI23	
X (49)	=	EI33	
X (50)	=	EI14	

X (51)	=	EI24
X (52)	=	EI34
X (53)	=	EI44
X (54)	=	EI15
X (55)	=	EI25
X (56)	=	EI35
X (57)	=	EI45
X (58)	=	EI55
X (59)	=	EI16
X (60)	=	EI26
X (61)	=	EI36
X (62)	=	EI46
X (63)	=	EI56
X (64)	=	EI66
X (65)	=	(Available)
.		.
.		.
.		.
X (75)	=	(Available)

NOTE D1(n) = Time Derivatives of X(n)

XI	temporary storage of X(n) in RKAD
Y	temporary storage for local use
YY	stored values of trajectory variables at beginning of coast
ZEH	$(\text{zeta})_H$
ZEH	$(\text{zeta})_M$
ZETA	true anomaly of position
ZETA2	ZETA at end of coast
ZM	Time from perigee to moon on parabolic trajectory

PROGRAM OPERATION

DATA INPUT FORMAT

Data Blocks

Data required for PRESTO execution are grouped and input in data blocks which are identified by number. Each block contains a common type of information such as the case title or a stage thrust table, and utilizes a specified FORTRAN format for the entire data block. The contents of each data block are discussed and then summarized on the next several pages.

Header Cards

Input data are punched on cards which are placed behind the program binary deck. Cards for each data block are preceded and identified by a "header" card. The format of the header cards is 4I3, where the first field is the data block number and the second and third fields give the (inclusive) locations within the data block that the subsequent data cards are to be placed. Thus, if the user wished to change constants 2, 3 and 4 in data block 10, the header card would be punched _10__2__4___, and the first three fields of the next card would contain the three constants. The fourth field of the header card is used to identify the vehicle stage number for the data, where necessary. The stage designation should be included only for the data blocks as specified on the following pages.

Card Sequencing

Cards within each data block must follow sequentially, but data blocks may be arranged in any order. A blank card must follow the last data card.

Successive Cases

Successive cases can be run on PRESTO with a minimum of additional data input. Only the changes in data from the preceding case must be input. A blank card must follow the data for each case.

999 Card

A card with 999 punched in the first three columns must end the data deck. It is to be placed behind the blank card which ends the data for the final case. When the READ routine encounters the 999 card a normal stop is indicated to the machine operator, so that the job can be terminated.

DISCUSSION OF THE DATA BLOCKS

Data Blocks 1, 2, 3

Alpha-numeric comment cards that are output at the top of each page. Data block 1 is the first line; data blocks 2 and 3 appear on the second line.

Data Block 4

Basic computation options as follows:

(1) Mission selection governs the form of the terminal constraints and the suggested input stage sequencing.

(2) Select only 3D axes options (three-dimensional point mass motion).

(3) In-plane control variable for initial nominal trajectory can be either eta (thrust angle to velocity vector) or theta (thrust angle to horizontal). Yaw thrust angle, chi, can be read from table if non-zero. The selected tables are read into data blocks 34, 35, 36.

(4) Eta and chi are both always optimized except when alpha is contrained to zero.

(5) The closed-form lift-off computation should always be used when the initial conditions correspond to ground lift-off. When selected, the first few seconds of motion are computed with closed-form expressions for velocity, altitude and mass. Vertical flight, sea level gravity and atmospheric pressure, and zero drag and lift forces are assumed. The time duration for this phase of boost is input in data block 20 and should be chosen to produce a velocity of approximately 200 feet/sec. The velocity, altitude and mass from this computation then become the initial conditions for numerical integration of the differential equations of motion, along with the flight path angle, azimuth, latitude and longitude from data block 21. (Zero altitude and velocity should be input to data block 21.) The purpose of this option is to avoid the low-velocity high-sensitivity region in numerical integration of azimuth and flight path angle.

(6) Vacuum thrust as a function of time from stage ignition is interpolated from input tables in data block 30 for each powered stage. Net thrust is computed as $T = T_V - p.A_e$ where p is the local ambient pressure and A_e is the engine nozzle exit area input to data block 39.

(7) Mass is computed within each stage as a function of time from the integration of $dm/dt = T_V/g_o Isp_V$. Thus, the vacuum specific impulse for each stage must be input in data block 39.

(8) Unnecessary aerodynamic force computations can be eliminated with this option to help speed the computation.

(9) A possible means of speed improvement in the future may be multiple usage of each integration of the adjoint equations. At present, these equations are solved after every successful forward trajectory.

(10) No special computations or print-out currently available. Interpolation to condition of maximum dynamic pressure is a possible future addition.

(11) Constraints on the control variables can be (a) on the product of dynamic pressure and total angle of attack or (b) constrain angle of attack to zero. The QBAR * ALPHA constraint is imposed as an inequality constraint. The theory and program mechanization for these constraints are discussed in Section 10. Further input data for use of these options is required in data block 22, including the time to release the constraint. This time should be input as a fraction of an integration step less than the actual desired time, due to the idiosyncrasies of the coding.

(12) Constraints on state variables during boost are concerned with maintaining the trajectory altitude above a specified minimum during upper-stage operation. Two techniques are available; both place inequality constraints on the altitude of coast stages. The more preferred of the two forces the trajectory to include a circular park orbit of some specified minimum altitude between two of the burns in stage 4. Use of this option is recommended for missions 2 and 5, and for mission 1 when boosting through a park orbit into transfer up to a higher terminal orbit. The second type of constraint available forces the perigee altitude of a (non-circular) coast above a specified value. Further description of these options, including the theory, mechanization and data input required, is given in Section 10.

(13) It is possible to constrain the total time from launch to final burnout on any mission for any configuration and stage sequence that includes a coast. The various aspects of this constraint are discussed in Section 10.

(14) Payload maximization can be conducted under either of two ground rules. In the first, final burnout weight is maximized while maintaining a fixed launch gross weight. Thus, the process is one of minimizing the final stage propellant consumption. In the second approach, the final stage propellant consumption is maintained at the input magnitude and the payload improvement is added to the launch gross weight. Thus, when using the fixed final stage option, the input initial and final weights of the final stage must accurately reflect the intended propellant consumption. For the fixed launch gross weight option, the stage burnout weights are not required. Further discussion of the fixed final stage option should be consulted as it appears in Section 10.

(15) For convenience, primarily in checkout, the floating-point numbers in common have been grouped and can be dumped in the floating-point format when requesting a dump at the end of the job.

(16) The version of this program delivered to Langley Research Center utilizes a non-buffered (FORTRAN II) output routine. A FAP-coded high speed output routine, using buffering of output, will be available for FORTRAN II usage in the near future.

(17) Selection of this option provides a three-page output of all the input data for each case. The data are identified by data block number and a few descriptive words.

Data Block 5

Specified sequence of stages, identified by number. Powered flight stages 1, 2, 3 have provision for aerodynamic characteristics. Powered flight stage 4 is to be used out of the atmosphere only. Stage 4 (only) can be split into one, two or three buring periods. Coast stage 5 is to be used when coasting in the atmosphere because the coast trajectory is computed using numerical integration. Aerodynamic characteristics are provided for this stage. Coast stages 6, 7, 8, 9 are computed with closed-form orbit equations and are, therefore, to be used only out of the atmosphere. Stage 8 is to be used only for the circular park orbit option (see Section 10), and stage 9 is to be used only for the coast perigee constraint (see Section 10).

The stage sequence must start with either stage 1 or stage 4. The closed-form lift-off computation and adjustable parameters 8, 9, 10 can be used only when the first stage is stage 1. Powered stages must appear only in increasing order; coast stages can appear in any order, with the one qualification that stage 8 must appear between two burns in stage 4.

For missions involving orbital initial conditions, where optimization of the ignition time for departure from orbit is to be accomplished, the following stage sequence should be used. Start with stage 1 for two integration steps, specifying zero thrust and initial conditions representing the desired initial orbit. Follow this with closed-form coast stage 6 as an adjustable parameter and end with the legitimate powered stage. Optimization of stage 6 duration is the optimization of the departure point.

Data Block 6

Specification of the adjustable parameter code numbers is based solely on the adjacent powered stage number. For each adjustable parameter selected a weighting constant must be input in data block 27. If optimizing the length of a coast which is expected to go to zero, specify a non-zero nominal coast angle; the program will eliminate the coast as an adjustable parameter if it is driven to zero.

Data Block 8

Specification of terminal constraint parameters. The nature of input to this data block is strongly dependent on the specified mission. For the Earth orbit mission a choice in stopping parameters is available Normally, the inertial velocity would be selected for convenience. However, in cases where inertial velocity does not vary monotonically near cutoff (usually due to transfer-orbit coasts), the total energy ($2E = V_I^2 - 2\mu/r$) should be used. The other terminal constraints are then listed in any order, with the one exception of altitude (or radius). There are indications that the matrix inversion routine retains higher accuracy if the altitude constraint (if selected at all) appears first in the list. The maximum allowable number of constraints for the program is six. Thus, if using the circular park orbit option (which imposes two constraints) a maximum of four constraints (including mass) can be specified in this data block.

For the lunar transfer missions, the transfer time to lunar radial distance constitutes the stopping parameter, and the right ascension and declination of vehicle position at the lunar distance are the nominal constraint parameters. Transfer -orbit inclination is an optional additional constraint. When this mission is specified, the lunar ephemeris routine provides the target right ascension and declination.

For the lunar descent mission, the input is identical with the orbit injection mission except that the local velocity should always be used for the stopping parameter.

The stopping parameter used for the planetary transfer missions is the magnitude of the hyperbolic excess velocity vector, which is equivalent to the square root of 2E. The right ascension and declination direction angles of this vector are the two constraint parameters. The user has the option of inputting the desired values of these three numbers in data block 19 or of simply specifying the target planet and the trip time. In the latter case the PLANEP subroutine automatically computes the required departure hyperbolic excess velocity vector. In both cases the launch date is a required input in data block 20.

Data Block 9

Frequency of output points. There is generally no need to output every computed point. In fact, from an economy standpoint, there is every need to minimize the output, since that operation is relatively slow under the FORTRAN II monitor. Output of nearly every computed point is often desirable on the final (optimum) trajectory, but can be almost entirely eliminated on the previous iterations. The user also has the ability to vary the output frequency between stages if, for example, a more frequent output is desired during the atmospheric phase of boost than is required for the upper stages. Regardless of the output frequency, the initial and final points of each stage are always output. There is no output of the trajectory variables on the backward runs.

Data Block 10

Nominal constants. Of particular note is the quantity HAERO, the altitude above which all aerodynamic computations are by-passed regardless of the aerodynamics option selected. This logic is in the interest of computation speed. The nominal altitude of 250,000 feet, for this cutoff, generally corresponds to a dynamic pressure of less than 1.0 lb/ft^2. All the constants in this data block are a part of the program binary deck and must be input as data only if a change is desired.

Data Blocks 11, 12, 13, 14

Constants for ATMOSphere subroutine. The 1959 ARDC model constitutes the nominal atmosphere as described in detail in Section 9. The various constants used in each exponential segment can be changed here with data input.

Data Block 15

Closed-form coast subroutine constants. As described in Section 8, discontinuities in the adjoint variables across coast stages are evaluated numerically using perturbations in the trajectory variables. The magnitudes of these perturbations can be controlled by data input, but there should now be no reason to input changes from the nominal.

Data Block 16

Control of stage times and number of integration steps. In order to gain maximum speed in integration, a constant time increment is used in each stage. The user specifies this integration frequency by inputting the stage duration and the number of integration steps desired for the stage. This information must be input for stages 1 through 5 if they have been specified in the stage sequence. The total number of integration steps must not exceed 150. Since all stages are nominally terminated on input stage time, special consideration must be given the final stage. In order to ensure that each forward trajectory ends when the desired value of the stopping parameter is reached, an artificially long burn duration must be input for the final stage. As a rough guide, extend the allowable burn sufficiently to provide a ten percent velocity margin. Specification of multiple burn in stage 4 requires the total stage duration, total number of points and the burnout time of each burn period expressed in burning time from ignition of the first burn.

Data Block 17

Various input data. The first five words of this data block must always be input. Words 3 through 15 contribute to an automatic computation of the initial mass improvement to be attempted. This computation is described in detail in Section 8. Word 16 is a limit on the adjustment in coast time in stage 5 on any one iteration and must be input if this stage is used as an adjustable parameter. This number should not exceed 100 seconds.

Data Block 19

Desired magnitudes of the terminal constraint parameters that were specified in data block 8. Note that since the list of constraints can appear in any order, the magnitudes must be in pure units; hence, all angles in radians.

Data Block 20

The second word in this data block is the launch date, expressed in decimal fractions of days since January 0.0, 1960. Expressed another way, it is the Julian date minus 2,436,934.5 days. It must be input whenever the terminal constraints involve an inertial longitude reference.

Data Block 21

Initial conditions on the trajectory variables are input in the rotating frame. This is convenient for boost problems; but for starting from orbit, one must generally be reminded that the velocity (e.g.) is not inertially referenced. Altitude is in feet above the planet surface, longitude in degrees ± from the prime meridian, latitude in degrees ± from the equator, azimuth in degrees east of north, flight path angle in degrees up from horizontal.

Data Block 22

Magnitudes for control variable constraints. Time to initiate and to release constraints should be input a fraction of an integration step before the actual desired time. These times are measured in seconds from launch, exclusive of coasts.

Data Block 23

Minimum allowable altitude for coast perigee constraint and circular park orbit constraint.

Data Block 24

Nominal range angles for closed-form coast stages. The independent variable used in orbit equations to describe the coast duration is the increment in true anomaly, or the central angle. This should be input as a non-zero angle for an adjustable coast.

Data Block 27

Weighting constants for optimization parameters. For numerical reasons, it is necessary to use non-unity weighting constants for some of the adjustable parameters. The recommended weighting constant for all closed-form coast stages is 1.0, for the integrated coast stage 5 use 10^{-5}, for adjustable burns use 10^{-3} (sometimes 10^{-4} is necessary), for the adjustable launch time of day use 10^{-9}, for the initial azimuth and flight path angle, use 2.0 and 1.0, respectively. Use 1.0 for the control variable chi. Smaller numbers will produce larger adjustments on any one iteration. The weighting constants must be input for all specified adjustable parameters and chi.

Data Block 28

Convergence data. The first word is the initial weight improvement (in pounds) to be used if the automatic computation of this number fails. Five percent of the expected burnout weight would be a reasonable number. The second word is the "epsilon weight" or allowable proximity to optimum for ceasing the optimization. When the attempted weight improvement becomes less than this number, the final guidance runs are made. Due to the halving process that is used and the usual non-linearities that are encountered, the user should input a number that is no bigger than one-third of the actual desired condition. For the third word, twenty iterations are generally more than sufficient. Fourth, the velocity below which the integration package is forced to stay in Runge-Kutta, should be no less than 1000 ft/sec. Further, if the vacuum thrust is time-variant within a stage, the Runge-Kutta mode should be used via this mechanism. The fifth word in this data block represents the specified point along the trajectory at which the switch between closed-loop and open-loop guidance is made. This is input as a velocity. Generally, the closed-loop computation is most effective when this velocity is approximately two thirds of the terminal velocity. However, if there is a coast stage of more than 100 seconds the open-loop computation should start before that coast.

Data Block 29

Allowable deviations in terminal and park orbit constraint parameters. Since each forward run is judged successful or not partially on the basis of satisfying the terminal constraints, an acceptable "dead band" must be specified for each constraint parameter. This "dead band"is important primarily in the intermediate iterations and hardly affects the constraints miss on the final (optimized) trajectory. If too tight limits are imposed, the mass improvement will proceed with unnecessarily small steps. If the limits are too loose, some iterations will unwisely be judged "successful." Recommended limits are 10,000 feet on distances and one degree on angles.

Data Block 30

Vacuum thrust tables are input as functions of stage time for each powered stage specified.

Data Block 31, 32, 33

Aerodynamic force coefficients are input as functions of Mach number for stages 1, 2, 3, 5 (not 4) if specified. The zero-lift drag coefficient is input unless options for zero aerodynamics are selected. The lift coefficient used is a $\partial C_L/\partial\alpha$ versus Mach number. The drag rise due to angle of attack is a $\partial C_D/\partial\alpha^2$ versus Mach number. The latter two tables must be input only if the option for aerodynamic lift is selected.

Data Blocks 34, 35, 36

Nominal pitch programs on theta, eta and chi. The use of these tables depends entirely on option 3.

Data Block 39

Stage data for stages 1 through 5 must be input for each stage specified in the stage sequence.

DATA INPUT

DATA BLOCK NUMBER	TITLE	FORMAT	COMMENTS
1	Main Heading	12A6	
2	Case Heading	12A6	
3	Date	2A6	
4	Options	24I1	IP(n) - - See list of available options
5	Stage Sequence	12I1	ISTGE(n) Powered Flight Stages 1 through 4 Atmospheric Coast Stage 5 Closed-form Coast Stages 6 through 9 * End Sequence with Zero
6	Adjustable Parameters	11I3	IB1(n) IB1(1)=The number of adjustable parameters to be optimized IB1(2)=The adjustable parameter : codes in non-decreasing IB1(11)order

** Adjustable Parameter Codes **

1 Length of Coast After Middle Burn in Stage 4
2 Length of Middle Burn in Stage 4
3 Length of Coast After First Burn in Stage 4
4 Length of First Burn in Stage 4
5 Length of Coast After Stage 3
6 Length of Coast After Stage 2
7 Length of Coast After Stage 1
8 Launch Time of Day
9 Initial Azimuth
10 Initial Flight Path Angle

8	Terminal Constraints	7I3	IB3(n) - - Mission-Dependent IB3(1) = Stopping Parameter Code IB3(2) = Form or Number of Terminal Constraints IB3(3) = List of Constraint Codes : for Orbit-injection and IB3(7) Lunar Descent Missions

** Stopping Parameter Codes **

A. Orbit-Injection Mission

1 TWOE ($V_I^2 - 2\mu/R$)
2 Available
3 Available
4 Inertial Velocity (V_I)
5 Available
6 Velocity

B. Lunar Transfer Mission

1 TransferTime to Lunar Radial Distance

** Stopping Parameter Codes **

C. Lunar Descent Mission

Use only Code 6 (Velocity)

D. Planetary Transfer Mission

1 Hyperbolic Excess Velocity (VH)

** Form or Number of Terminal Constraints **

A. Orbit-Injection Mission

and

C. Lunar Descent Mission

The Number of Terminal Constraints (1 through 6)

* Include the Terminal Mass Constraint in this Count

B. Lunar Transfer Mission

1 Use Ephemeris, Do Not Constrain Inclination of Transfer Orbit
2 Use Ephemeris, Do Constrain Inclination

D. Planetary Transfer Mission

1 Do Not Use Ephemeris (Input VH Vector)
2 Mars
3 Venus

** Constraint Codes For Orbit-Injection and Lunar Descent Missions **

1 Perigee Radius
2 Orbit Inclination
3 Longitude of Ascending Node
4 Argument of Perigee
5
6 Altitude
7 Inertial Flight Path Angle
8 Longitude
9 Inertial Azimuth
10 Latitude
11 Available
12 Available
13 Available
14 Radius

DATA INPUT

DATA BLOCK NUMBER	TITLE	FORMAT	COMMENTS
9	Output Frequency	24I3	IB4(n) Frequency Defined As: $\frac{\text{(number of computed points)}}{\text{(number of output points)}}$ IB4(1) - Initial Trajectory IB4(2) - Intermediate IB4(3) - Final IB4(4) - Stage Code IB4(5) - Initial Trajectory IB4(6) - Intermediate IB4(7) - Final IB4(8) - Stage Code IB4(9) - Initial Trajectory IB4(10) - Intermediate IB4(11) - Final IB4(12) - Stage Code IB4(13) - Initial Trajectory IB4(14) - Intermediate IB4(15) - Final IB4(16) - Stage Code IB4(17) - Initial Trajectory IB4(18) - Intermediate IB4(19) - Final IB4(20) - Stage Code
10	Nominal Constants	11E12.8	CT1(n) Nominal Values Shown Input only if changes are desired CT1(1) = 7.29211E-5 Rad/Sec Omega CT1(2) = 1.407735E16 Ft^3/Sec^2 EMU CT1(3) = 20902900. Ft RE CT1(4) = 1716.4827 Gas constant for atmosphere subroutine CT1(5) = 32.154856 Ft/Sec^2 measured sea level gravity; weight-to-mass conversion factor (GO) CT1(6) = 2116.2 lb/Ft^2 PO sea level ambient pressure CT1(7) = 250,000. Ft HAERO Altitude above which all aerodynamic computations are by-passed CT1(8) = 46. Upper limit of lines per page CT1(9) = (Available) CT1(10) = .02 sec TIMEP Time-epsilon increment to insure hitting critical time

DATA INPUT

DATA BLOCK NUMBER	TITLE	FORMAT	COMMENTS
11	Atmosphere	8E12.8	CT2(n) Nominal values tabulated
12	Subroutine	8E12.8	CT3(n) in report section
13	Constants	8E12.8	CT4(n) describing atmosphere
14	"	8E12.8	CT5(n) subroutine * Input only if changes from nominal values are desired
15	Nominal Constants for COAST Subroutine	8E12.8	CT6(n) * Input only for Change. Used in forming partial derivatives across closed-form coasts. CT6(1) = 1.0 Velocity Increment CT6(2) = .001 Gamma Increment CT6(3) = 100. Radius Increment CT6(4) = .001 Tau Increment CT6(5) = .1 Mass Increment CT6(6) = .001 Psi Increment CT6(7) = .001 Lambda Increment CT6(8) = .001 Coast angle Increment
16	Time Duration and Number of Integration Steps Within Each Stage	12E12.8	DA1(n) Time in seconds DA1(1) Stage 1 Time Duration DA1(2) Stage 1 Points DA1(3) Stage 2 Time DA1(4) Stage 2 Points DA1(5) Stage 3 Time DA1(6) Stage 3 Points DA1(7) Stage 4 Total Time Duration DA1(8) Stage 4 Total Points DA1(9) Stage 4 First Burnout Time DA1(10) Stage 4 Second Burnout Time DA1(11) Stage 5 Time Duration(nominal) DA1(12) Stage 5 Points
17	Input Data Used in Calculation for Initial Mass Improvement	20E12.8	DA2(n) DA2(1) Input should = GO DA2(2) = +1.0 for Take-off = -1.0 for Retro-burns DA2(3) Total Expected Burn Time (TIMT) DA2(4) Nominal delta eta * DA2(5) Nominal delta chi * * Suggest .05 (radians) DA2(6) Nominal delta tau 1. through : delta tau 10. DA2(15) DA2(16) Delta tau limit periteration on Coast Stage 5 DA2(17) Magnitude for Total Time Constraint. DA2(18) Minimum Coast Angle for Closed-Form Coast Stages

DATA INPUT

DATA BLOCK NUMBER	TITLE	FORMAT	COMMENTS
19	Terminal Constraint Magnitudes	6E12.8	DA4(n) Mission-Dependent DA4(1) is the magnitude of the stopping parameter DA4(2) : DA4(6) are the desired values of the terminal constraints, listed in the same order as in Data Block Number 8 * Units are feet, radians, seconds

** Terminal Constraints Input Guide **

A. and C. Orbit-Injection and Lunar Descent Missions

As Described Above

B. Lunar Transfer Mission

DA4(1) Transfer Time (hours)
DA4(2) DA4(3) Used Internally
DA4(4) Inclination of Transfer Orbit (radians)

D. Planetary Transfer Mission

DA4(1) Hyperbolic Excess Velocity
DA4(2) Right Ascension of Hyperbolic Asymptote (radians)
DA4(3) Declination of Hyperbolic Asymptote (radians)
DA4(6) Transfer Time (days) (DA4(4) and DA4(5) are used internally)
* DA4(1,2,3) are not required input if PLANEP routine is called

DATA BLOCK NUMBER	TITLE	FORMAT	COMMENTS
20	Initial Times	2E12.8	DA5(n) DA5(1) Time Duration for Closed-form Lift-off Calculation (sec.) DA5(2) Launch Time in Space Age Date (days from January 0.0, 1960)
21	Initial Conditions	6E12.8	DA6(n) DA6(1) Altitude (feet) DA6(2) Longitude (degrees ± from Prime meridian) DA6(3) Latitude (degrees) DA6(4) Velocity (feet/second) DA6(5) Azimuth (degrees) DA6(6) Flight path angle (degrees)
22	Control Variable Constraints	6E12.8	DA7(n) DA7(1) QBAR*ALPHA Magnitude (lb/ft^2)*(radians) DA7(2) Time to Start Testing DA7(3) Time to Release Constraint For α = 0 Constraint DA7(5) Time to Start Testing DA7(6) Time to Release Constraint

DATA INPUT

DATA BLOCK NUMBER	TITLE	FORMAT	COMMENTS
23	Constraints on State Variables During Boost	4E12.8	DA8(n) DA8(1) Available DA8(2) Available DA8(3) Minimum Altitude for Coast Stages 8 and 9 (feet) DA8(4) Available
24	Nominal Range Angles for Closed-Form Coast Stages	4E12.8	DA9(n) Angles In radians of Central Arc DA9(1) Coast Stage 6 DA9(2) Coast Stage 7 DA9(3) Coast Stage 8 DA9(4) Coast Stage 9
27	Weighting Constants For Optimization Parameters	11E12,8	DA12(n) DA12(1) ⋮ DA12(10) For Adjustable parameters identified by code number as listed for DATA Block 6 DA12(11) For Control Variable Chi
28	Convergence Data	6E12.8	DA13(n) DA13(1) Initial Weight Improvement to be used if (automatic) internal computation fails DA13(2) Epsilon-weight for stopping attempted mass improvement DA13(3) Maximum Number of Forward Trajectories per case DA13(4) Velocity below which Runge Kutta integration is always used. DA13(5) Velocity to start open-loop computations.
29	Permitted Values of Terminal Constraint and coast perigee constraint deviations	5E12.8	DA14(1) ⋮ DA14(5) List in same order as constraints are specified and in the same units * Suggest 10,000 feet for distances and 1 degree on angles
30	Vacuum Thrust Tables	12E12.8	TT1,TT2,TT3,TT4 * Stage Code Required on Header Cards Sequence: Blank, time, thrust, time, thrust....... 1.0E10
31	Drag Coefficient Tables (ALPHA = 0.)	22E12.8	TD1,TD2,TD3,TD5 * Stage Code Required on Header Cards Sequence: Blank, Mach, C_D, Mach, C_D, Mach.....1.0E10

DATA INPUT

DATA BLOCK NUMBER	TITLE	FORMAT	COMMENTS
32	Lift Coefficient per ALPHA Tables	22E12.8	TL1,TL2,TL3,TL5 * Stage Code Required on Header Cards Same Sequence: Mach vs. C_{L_α} Units: C_L per radian
33	K-Drag Tables (Drag Due to Non-Zero ALPHA)	22E12.8	TK1,TK2,TK3,TK5 * Stage Code Required on Header Cards Same Sequence: Mach vs. $K_{D_{\alpha^2}}$ Units: K_D per radian^2
34	Theta History for Nominal Trajectory	32E12.8	TTH Same Sequence: Theta vs. time Units: degrees, time from launch exclusive of coasts
35	Eta History for Nominal Trajectory	32E12.8	TET Same Sequence: Eta vs. time Units: degrees, time from launch exclusive of coasts
36	Chi History for Nominal Trajectory	32 E12.8	TCH Same Sequence: Chi vs. time Units: degrees, time from launch exclusive of coasts
39	Stage Data	5E12.8	SG1,SG2,SG3,SG4,SG5 each dimensioned 5 * Stage Code Required on Header Cards SGx(1) Vacuum I_{sp} (sec) SGx(2) Aerodynamic Reference Area (ft^2) SGx(3) Total Nozzle Exit Area (ft^2) SGx(4) Initial Weight (lb) SGx(5) Final Weight (lb)

* NOTE: When Inputting Changes to Theta, Eta, or Chi tables the entire table must be re-input.

DATA INPUT

** LIST OF AVAILABLE OPTIONS FOR DATA BLOCK NUMBER 4 **

IP(1)	Mission	1	Orbit-Injection
		2	Inject into Lunar Transfer from Earth Lift-off
		3	Inject into Lunar Transfer from Earth Orbit
		4	Lunar Landing
		5	Inject into Planetary Transfer from Earth Lift-off
		6	Inject into Planetary Transfer from Earth Orbit
IP(2)	Axes	0	Rotating and 3D
		1	Rotating and 2D
		2	Non-rotating and 3D
		3	Non-rotating and 2D
IP(3)	Control Variables For Nominal Trajectory	0	Theta from Input Table and Chi = 0.
		1	Theta and Chi from Input Tables
		2	Eta from Input Table and Chi = 0.
		3	Eta and Chi from Input Tables
IP(4)	Control Variables to be Optimized	0	Eta only
		1	Eta and Chi
		2	Chi only
IP(5)	Closed-Form Lift-off Computation	0	Use this Computation
		1	Do not use this Computation
IP(6)	Thrust	0	Table Look-up - - Vacuum Thrust vs. time
IP(7)	Mass Computation	0	MDOT = $T_V/(G_0*I_{sp})$
IP(8)	Aerodynamics	0	Lift = 0. only
		1	Drag = 0. only
		2	Lift=Drag= 0.
		3	Include Lift and Drag
IP(9)	Frequency of New Adjoint Solution	1	After every successful forward trajectory
IP(10)	Special Computations and Print-out	0	None
IP(11)	Constraints on Control Variables	0	None
		1	QBAR*ALPHA
		2	ALPHA = 0.0
IP(12)	Constraints on State Variables During Boost	0	None
IP(13)	Total Time Contraint	0	Do not constrain total time
		1	Do constrain total time
IP(14)	Mass-Improvement Procedure	0	Fixed launch weight
		1	Fixed Final Stage

** LIST OF AVAILABLE OPTIONS **

IP(15)	Memory Dump	0	Do not dump memory
		1	Dump memory at end of job (includes floating-point dump of floating-point numbers in common)
IP(16)	Output	0	Buffered Output
		1	Non-Buffered Output
IP(17)	Output of Input Data	0	Do not output data
		1	Output data

OUTPUT FORMAT

The output of PRESTO consists primarily of the tabulated history of the trajectory variables for each forward iteration. There is also an output of all data which has been input (selected by option 17), output from the lunar and planetary ephemeris subroutines if they are used, output of the corrections to be made in the terminal constraints on each iteration, and output of several mass improvement parameters. A definition of all output quantities follows.

Column Headings for Trajectory Listings

TIME	Total time from launch
VEL	Velocity in rotating frame
GAMMA	Vertical-plane path angle of VEL from horizontal
PSI	Azimuth of VEL, degrees east of north
ALTITUDE	Distance above planet surface
LAM	Latitude, degrees north of equator
TAU	Longitude in rotating frame, degrees east of prime meridian
QBAR	Dynamic pressure
THRUST	Net thrust
WEIGHT	Sea level weight
THETA	Thrust angle to horizontal
CHI	Yaw thrust angle of attack
ETA	Vertical plane thrust angle of attack
DCHI	Change in CHI from current nominal trajectory
DETA	Change in ETA from current nominal trajectory
VI	Velocity in inertial frame
GAMI	Vertical-plane path angle of VI
PSII	Azimuth of VI
MACH	Mach number, VEL/local speed of sound
ALPHA	Total angle between THRUST and VEL
DRAG	Total aerodynamic drag
LIFT	Aerodynamic lift
QALPHA	Product of QBAR and ALPHA
PSIBAR	Azimuth of thrust vector

Output for Closed-Form Coast Stages

TWOE	$2E = V_I^2 - 2\mu/r$
EH	Angular momentum, $V_I \cdot r \cdot \cos \gamma_I$
RP	Perigee radius
EYE	Inclination, radians
BETAC	Central angle of coast, radians

Output of Terminal Conditions Orbit Elements

TWOE, EH, RP	Same as Closed-Form Coast
EYE	Inclination, degrees
BETAP	Argument of perigee, degrees
OMEGAE	Longitude of ascending node, degrees

Output at Start of Each Iteration

Terminal Constraint Corrections — These quantities, called $d\psi_i$ in the equations, are the negative of the errors in terminal constraints on the current nominal. Units are feet and radians.

DWFINAL	Improvement in final weight attempted on each iteration
DWFUEL	See discussion of Fixed Final Stage Option
DWPAYLOAD	in Section 10
DMASD	Initial weight improvement attempt
LAUNCH TIME	Space Age Date at start of trajectory

Special Output for Lunar Transfer Missions

(a) Output of lunar position at nominal arrival time and one-half day later. This is initial output for case, computed from lunar ephemeris subroutine.

ALPHA	Right ascension of lunar position, degrees
DELTA	Declination of lunar position, degrees
DISTANCE	Radial distance from Earth center to Moon center at nominal arrival time

(b) Terminal Conditions

TM	Transfer time to Moon, hours
ALFAM	Computed vehicle right ascension at Moon
DELTAM	Computed vehicle declination at Moon
EYE	Inclination of transfer orbit

(c) Updated time of arrival and corresponding lunar position are computed after each iteration, taking into account the adjusted launch time, coast durations, etc.

Special Output for Planetary Transfer Missions

(a) Initial output of required hyperbolic excess velocity vector magnitude, and right ascension and declination angles computed from PLANEP subroutine.

(b) Terminal Conditions

VH ALFAP DELTAP	Departure hyperbolic asymptote computed from terminal trajectory variables

TROUBLE SHOOTING

Listed in this section are a few of the more common difficulties which can be encountered in the use of PRESTO. Most of the troubles result from omitted or wrong data input. The remainder of cases involve the timing of the switch from closed-loop to open-loop computation. This point will be referrred to as T9. The symptoms are listed along with the probable cause.

- No print out of final line of output giving orbit elements at end of each forward trajectory.

  Reason: Insufficient burn time allowed in final stage.

- Failure to converge properly in guidance iterations.

  Reason: (1) Erroneous specification of terminal constraints.

  (2) Placement of T9 after an extended coast.

- Runs only guidance trajectories without starting optimization.

  Reason: Failure to input allowable deviations in Data Block 29.

- Failure to meet terminal constraints well while optimizing.

  Reason: T9 is too early in the trajectory.

- DETA saturating at end of trajectory (printing out ±15.01 degrees).

  Reason: T9 is too late in the trajectory.

- Zero increment computed for adjustable parameters.

  Reason: Weighting constants have not been input.

- Unduly large increment computed for adjustable parameter.

  Reason: Weighting constant is too small.

- Zero increment computed for adjustable coast stage 5.

  Reason: Failure to input DA2 (16) in data block 17.

- Zero DCHI on any rotating-Earth or out-of-plane boost.

  Reason: Weighting constant for CHI has not been input to data block 27.

EQUATIONS

LIST OF SYMBOLS

A	Aerodynamic reference area
sign A_1	+1 for takeoffs and -1 for retroburns
A_e	Rocket nozzle exit area
a	Local speed of sound
C_D	Aerodynamic drag coefficient
C_L	Aerodynamic lift coefficient
D	Aerodynamic drag force
E	Total energy of orbit (or coast)
F	dV/dt
G	$d\gamma/dt$
g	Local acceleration due to gravity
g_e	Sea level g
H	$d\psi/dt$
H	Angular momentum of orbit
h	Altitude above planet surface
I	dr/dt
i	Inclination of orbit plane to equator
J'	$d\lambda/dt$
K	$d\tau/dt$
L	Aerodynamic lift force
M	Mach number
m	Mass of vehicle
p	Local atmospheric pressure
r	Radial distance from planet center to vehicle
r_p	Perigee radius of orbit
r_α	Semi-major axis of orbit
r_M	Lunar radial distance from Earth-center
T	Net thrust force
T_G	Space Age Date = Julian date - 2,436,934.5
t	Time from pericenter of orbit
T_V	Thrust measured in a vacuum
V	Velocity in rotating frame
V_I	Velocity in inertial frame

α	Angle-of-Attack
α_e	Right ascension
α_{e_H}	Right ascension of the hyperbolic asymptote
α_{e_M}	Right ascension of the vehicle at the Moon's radius
β	In-plane range angle from the ascending node
β_c	Range angle of coast
β_H	Angle between ascending node and hyperbolic asymptote
β_M	In-plane range angle at the Moon's radius from the ascending node
β_p	Argument of perigee
γ	Angle between V and local horizontal (flight path angle)
γ_I	Angle between V_I and local horizontal (inertial flight path angle)
δ_e	Declination
δ_{e_H}	Declination of the hyperbolic asymptote
δ_{e_M}	Declination of the vehicle at the Moon's radius
ϑ	True anomaly
ϑ_H	Limiting value of true anomaly of hyperbola
ϑ_M	True anomaly at Moon's radius
η	Angle between thrust vector and $\bar{i}_x - \bar{i}_y$ plane (see page 7-2)
θ	Angle between thrust vector and local horizontal
λ	Latitude
μ	Gravity constant
ν	Inertial longitude angle from ascending node
ν_H	Inertial longitude of hyperbolic asymptote
ν_M	Inertial longitude of the vehicle at the Moon's radius
ρ	Atmospheric density
τ	Longitude with respect to the rotating Earth
ϕ	Angle between lift vector and $\bar{i}_z$ axis (see page 7-2)
χ	Angle between $\bar{i}_y$ axis and the projection of the thrust vector on the $\bar{i}_x - \bar{i}_y$ plane (see page 7-2)
ψ	Angle between North and the horizontal component of V (azimuth)
ψ_I	Angle between North and the horizontal component of V_I (inertial azimuth)
ω	Earth rotation rate
Ω_e	Longitude of ascending node

3-D EQUATIONS OF MOTION

ROTATING COORDINATE SYSTEM

$$F = \dot{V} = r\omega^2\left[\cos^2\lambda\sin\gamma - \sin\lambda\cos\lambda\cos\psi\cos\gamma\right] - g\sin\gamma + \frac{T(t)}{m}\cos\eta\cos\chi\,\mathrm{sgn}\,A_1 - \frac{D(t)}{m}$$

$$G = \dot{\gamma} = 2\omega\cos\lambda\sin\psi + \frac{V}{r}\cos\gamma + \frac{r\omega^2}{V}\left[\cos^2\lambda\cos\gamma + \sin\lambda\cos\lambda\cos\psi\sin\gamma\right] - \frac{g}{V}\cos\gamma + \frac{T}{mV}\sin\eta\,\mathrm{sgn}\,A_1 + \frac{L}{mV}\cos\phi$$

$$H = \dot{\psi} = \frac{V}{r}\,\frac{\cos\gamma\sin\psi\sin\lambda}{\cos\lambda} + 2\omega\sin\lambda + \frac{r\omega^2\sin\lambda\cos\lambda\sin\psi}{V\cos\gamma} - \frac{2\omega\cos\lambda\cos\psi\sin\gamma}{\cos\gamma} + \frac{T\cos\eta\sin\chi}{mV\cos\gamma}\,\mathrm{sgn}\,A_1 + \frac{L\sin\phi}{mV\cos\gamma}$$

$$I = \dot{r} = V\sin\gamma$$

$$J = \dot{\lambda} = \frac{V\cos\gamma\cos\psi}{r}$$

$$K = \dot{\tau} = \frac{V\cos\gamma\sin\psi}{r\cos\lambda}$$

$$\dot{m} = \frac{-T_V}{g_e I_{sp}}$$

WHERE

$$g = \frac{\mu}{r^2} \qquad L = \tfrac{1}{2}\rho V^2 C_L A$$

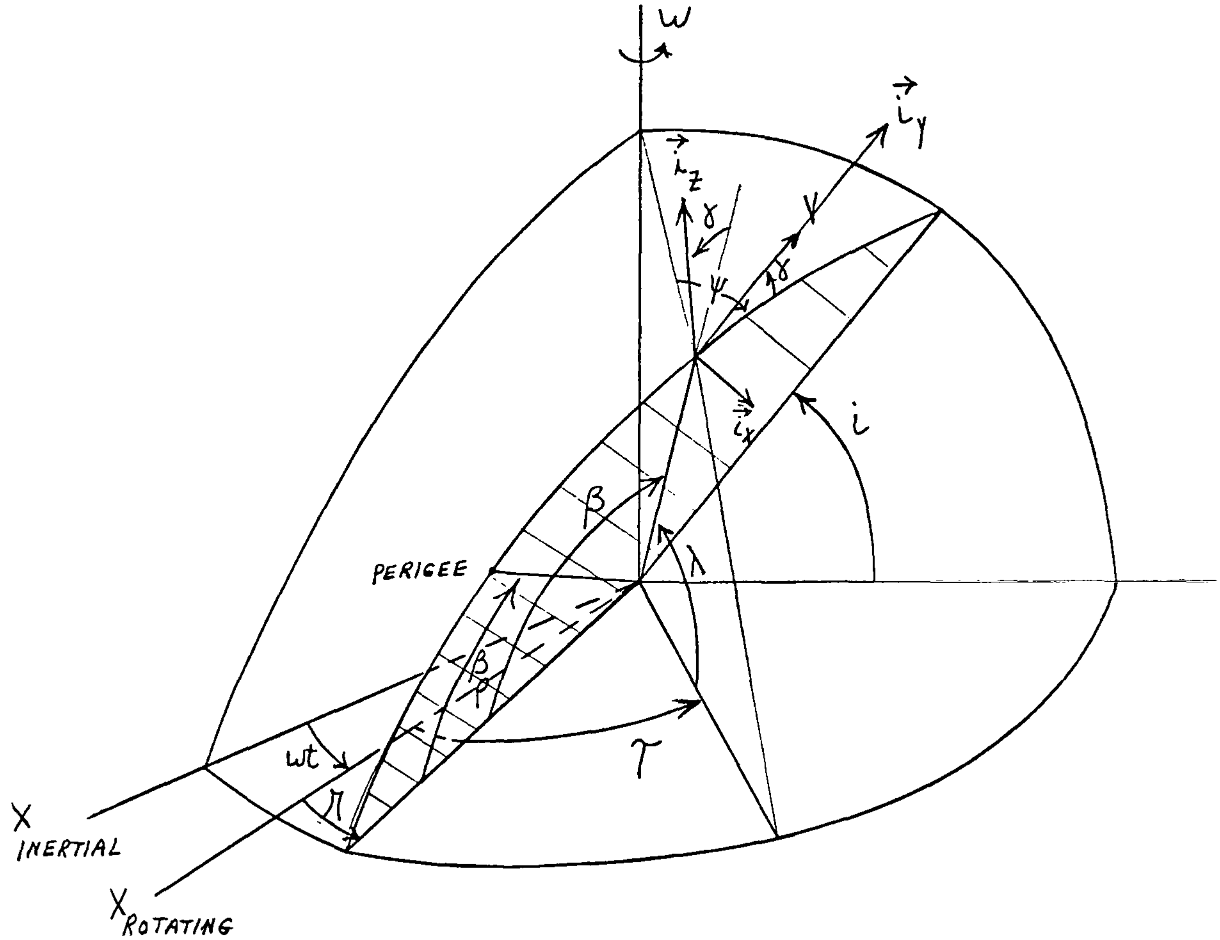
$$T = T_V - pA_e$$

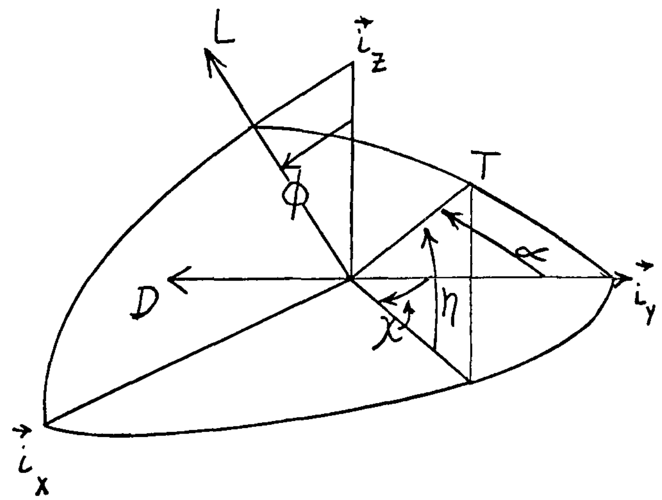
$\mathrm{sgn}\,A = +1$ FOR TAKEOFFS
$= -1$ FOR RETROBURNS

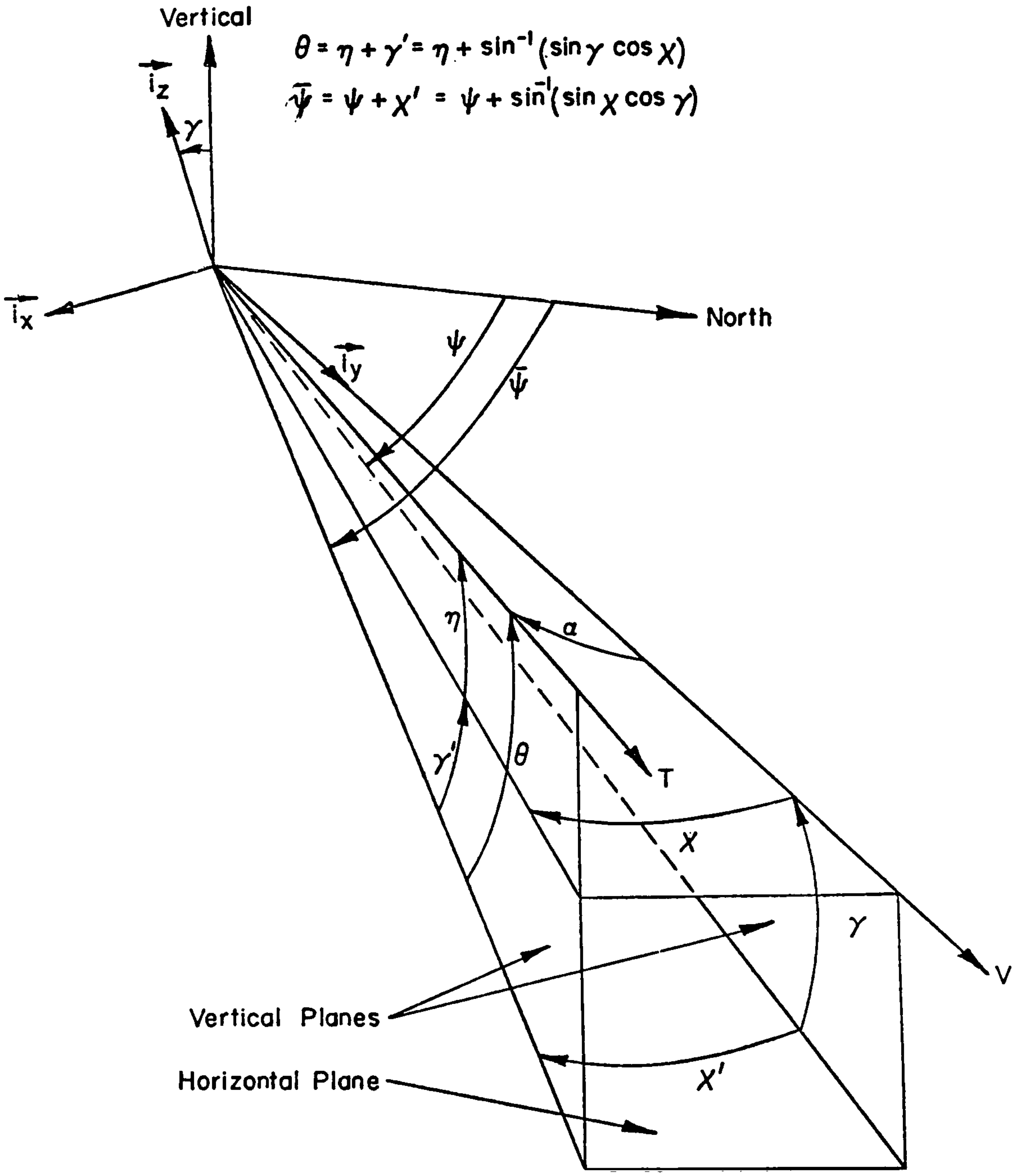
$$D = \tfrac{1}{2}\rho V^2 C_D A$$

η AND χ ARE CONTROL VARIABLES

DIAGRAMS OF TRAJECTORY VARIABLES AND FORCE ORIENTATION ANGLES







Control Angles Geometry

$$\sin\phi = \frac{\sin\chi}{\sqrt{\sin^2\chi + \dfrac{\sin^2\eta}{\cos^2\eta}}} \qquad \cos\phi = \frac{\sin\eta / \cos\eta}{\sqrt{\sin^2\chi + \dfrac{\sin^2\eta}{\cos^2\eta}}}$$

ANGLE-OF-ATTACK $\quad \alpha = \cos^{-1}(\cos\chi\cos\eta)\,\mathrm{sgn}\,\eta$

p = PRESSURE IS A FUNCTION OF $h = r - r_0$

ρ = DENSITY IS A FUNCTION OF h

C_D, C_L ARE FUNCTIONS OF α, M (MACH NUMBER)

$M = \frac{V}{a}$ WHERE a IS A FUNCTION OF h

LINEAR PERTURBATION EQUATIONS

$$\frac{d\delta V}{dt} = \frac{\partial F}{\partial V}\delta V + \frac{\partial F}{\partial \gamma}\delta\gamma + \frac{\partial F}{\partial \psi}\delta\psi + \frac{\partial F}{\partial r}\delta r + \frac{\partial F}{\partial \lambda}\delta\lambda + \frac{\partial F}{\partial m}\delta m + \frac{\partial F}{\partial \eta}\delta\eta + \frac{\partial F}{\partial \chi}\delta\chi$$

$$\frac{d\delta \gamma}{dt} = \frac{\partial G}{\partial V}\delta V + \frac{\partial G}{\partial \gamma}\delta\gamma + \frac{\partial G}{\partial \psi}\delta\psi + \frac{\partial G}{\partial r}\delta r + \frac{\partial G}{\partial \lambda}\delta\lambda + \frac{\partial G}{\partial m}\delta m + \frac{\partial G}{\partial \eta}\delta\eta + \frac{\partial G}{\partial \chi}\delta\chi$$

$$\frac{d\delta \psi}{dt} = \frac{\partial H}{\partial V}\delta V + \frac{\partial H}{\partial \gamma}\delta\gamma + \frac{\partial H}{\partial \psi}\delta\psi + \frac{\partial H}{\partial r}\delta r + \frac{\partial H}{\partial \lambda}\delta\lambda + \frac{\partial H}{\partial m}\delta m + \frac{\partial H}{\partial \eta}\delta\eta + \frac{\partial H}{\partial \chi}\delta\chi$$

$$\frac{d\delta r}{dt} = \frac{\partial I}{\partial V}\delta V + \frac{\partial I}{\partial \gamma}\delta\gamma \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0$$

$$\frac{d\delta \lambda}{dt} = \frac{\partial J}{\partial V}\delta V + \frac{\partial J}{\partial \gamma}\delta\gamma + \frac{\partial J}{\partial \psi}\delta\psi + \frac{\partial J}{\partial r}\delta r \quad 0 \quad 0 \quad 0 \quad 0$$

$$\frac{d\delta \Gamma}{dt} = \frac{\partial K}{\partial V}\delta V + \frac{\partial K}{\partial \gamma}\delta\gamma + \frac{\partial K}{\partial \psi}\delta\psi + \frac{\partial K}{\partial r}\delta r + \frac{\partial K}{\partial \lambda}\delta\lambda \quad 0 \quad 0 \quad 0$$

$$\frac{d\delta m}{dt} = 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0$$

Adjoint Equations

$$\frac{d\lambda_V}{dt} = -\lambda_V\frac{\partial F}{\partial V} - \lambda_\gamma\frac{\partial G}{\partial V} - \lambda_\psi\frac{\partial H}{\partial V} - \lambda_r\frac{\partial I}{\partial V} - \lambda_\lambda\frac{\partial J}{\partial V} - \lambda_\tau\frac{\partial K}{\partial V}$$

$$\frac{d\lambda_\gamma}{dt} = -\lambda_V\frac{\partial F}{\partial \gamma} - \lambda_\gamma\frac{\partial G}{\partial \gamma} - \lambda_\psi\frac{\partial H}{\partial \gamma} - \lambda_r\frac{\partial I}{\partial \gamma} - \lambda_\lambda\frac{\partial J}{\partial \gamma} - \lambda_\tau\frac{\partial K}{\partial \gamma}$$

$$\frac{d\lambda_\psi}{dt} = -\lambda_V\frac{\partial F}{\partial \psi} - \lambda_\gamma\frac{\partial G}{\partial \psi} - \lambda_\psi\frac{\partial H}{\partial \psi} \qquad - \lambda_\lambda\frac{\partial J}{\partial \psi} - \lambda_\tau\frac{\partial K}{\partial \psi}$$

$$\frac{d\lambda_r}{dt} = -\lambda_V\frac{\partial F}{\partial r} - \lambda_\gamma\frac{\partial G}{\partial r} - \lambda_\psi\frac{\partial H}{\partial r} \qquad - \lambda_\lambda\frac{\partial J}{\partial r} - \lambda_\tau\frac{\partial K}{\partial r}$$

$$\frac{d\lambda_\lambda}{dt} = -\lambda_V\frac{\partial F}{\partial \lambda} - \lambda_\gamma\frac{\partial G}{\partial \lambda} - \lambda_\psi\frac{\partial H}{\partial \lambda} \qquad - \lambda_\tau\frac{\partial K}{\partial r}$$

$$\frac{d\lambda_\tau}{dt} = 0$$

$$\frac{d\lambda_m}{dt} = -\lambda_V\frac{\partial F}{\partial m} - \lambda_\gamma\frac{\partial G}{\partial m} - \lambda_\psi\frac{\partial H}{\partial m}$$

Evaluation of Partial Derivatives

$$\frac{\partial F}{\partial V} = -\frac{1}{m}\rho V A\left[C_D + \frac{1}{2}\frac{V}{a}\frac{\partial C_D}{\partial M}\right]$$

$$\frac{\partial F}{\partial \gamma} = r\omega^2\left[\cos^2\lambda\cos\gamma + \sin\lambda\cos\lambda\cos\psi\sin\gamma\right] - g\cos\gamma$$

$$\frac{\partial F}{\partial \psi} = r\omega^2\sin\lambda\cos\lambda\sin\psi\cos\gamma$$

$$\frac{\partial F}{\partial r} = \omega^2\left[\cos^2\lambda\sin\gamma - \sin\lambda\cos\lambda\cos\psi\cos\gamma\right] + \frac{2\mu}{r^3}\sin\gamma - \frac{A_e}{m}\cos\eta\cos\chi\,\mathrm{sgn}\,A_1\frac{dp}{dh}$$
$$+ \frac{V^2A}{2m}\left[\rho\frac{V}{a^2}\frac{da}{dh}\frac{\partial C_D}{\partial M} - C_D\frac{d\rho}{dh}\right]$$

$$\frac{\partial F}{\partial \lambda} = r\omega^2\left[-2\cos\lambda\sin\lambda\sin\gamma - \cos^2\lambda\cos\psi\cos\gamma + \sin^2\lambda\cos\psi\cos\gamma\right]$$

$$\frac{\partial F}{\partial m} = -\frac{T}{m^2}\cos\eta\cos\chi\,\mathrm{sgn}\,A_1 + \frac{D}{m^2}$$

$$\frac{\partial F}{\partial \eta} = -\frac{T}{m}\sin\eta\cos\chi\,\mathrm{sgn}\,A_1 - \frac{\rho V^2A}{2m}\cos\phi\cos\chi\frac{\partial C_D}{\partial \alpha}$$

$$\frac{\partial F}{\partial \chi} = -\frac{T}{m}\cos\eta\sin\chi\,\mathrm{sgn}\,A_1 - \frac{\rho V^2A}{2m}\sin\phi\frac{\partial C_D}{\partial \alpha}$$

$$\frac{\partial G}{\partial V} = \frac{\cos\gamma}{r} - \frac{r\omega^2}{V^2}\left[\cos^2\lambda\cos\gamma + \sin\lambda\cos\lambda\cos\psi\sin\gamma\right] + \frac{g}{V^2}\cos\gamma$$
$$-\frac{T}{mV^2}\sin\eta\,\mathrm{sgn}\,A_1 + \frac{\rho A\cos\phi}{2m}\left[C_L + \frac{V}{a}\frac{\partial C_L}{\partial M}\right]$$

$$\frac{\partial G}{\partial \gamma} = -\frac{V}{r}\sin\gamma + \frac{r\omega^2}{V}\left[\sin\lambda\cos\lambda\cos\psi\cos\gamma - \cos^2\lambda\sin\gamma\right] + \frac{g}{V}\sin\gamma$$

$$\frac{\partial G}{\partial \psi} = 2\omega \cos\lambda \cos\psi - \frac{r\omega^2}{V}\sin\lambda\cos\lambda\sin\psi\sin\gamma$$

$$\frac{\partial G}{\partial r} = -\frac{V}{r^2}\cos\gamma + \frac{\omega^2}{V}\left[\cos^2\lambda\cos\gamma + \sin\lambda\cos\lambda\cos\psi\sin\gamma\right] + \frac{2\mu}{r^3 V}\cos\gamma$$
$$-\frac{1}{mV}\sin\eta\, A_e\frac{dp}{dh}\operatorname{sgn}A_1 + \frac{VA}{2m}\left[C_L\frac{d\rho}{dh} - \rho\frac{V}{a^2}\frac{da}{dh}\frac{\partial C_L}{\partial M}\right]\cos\phi$$

$$\frac{\partial G}{\partial \lambda} = -2\omega\sin\lambda\sin\psi + \frac{r\omega^2}{V}\left[-2\cos\lambda\sin\lambda\cos\gamma + \cos^2\lambda\cos\psi\sin\gamma - \sin^2\lambda\cos\psi\sin\gamma\right]$$

$$\frac{\partial G}{\partial m} = -\frac{T}{m^2 V}\sin\eta\operatorname{sgn}A_1 - \frac{L}{m^2 V}\cos\phi$$

$$\frac{\partial G}{\partial \eta} = \frac{T}{mV}\cos\eta\operatorname{sgn}A_1 + \frac{\rho VA}{2m}\left[\frac{\partial C_L}{\partial \alpha}\cos^2\phi\cos\chi + C_L\frac{\sin^3\phi}{\sin\chi\cos^2\eta}\right]$$

$$\frac{\partial G}{\partial \chi} = \frac{\rho VA}{2m}\left[\frac{\partial C_L}{\partial \alpha}\sin\phi\cos\phi - C_L\frac{\cos\phi\sin^2\phi\cos\chi}{\sin\chi}\right]$$

$$\frac{\partial H}{\partial V} = \frac{\cos\gamma\sin\psi\sin\lambda}{r\cos\lambda} - \frac{r\omega^2\sin\lambda\cos\lambda\sin\psi}{V^2\cos\gamma} - \frac{T\cos\eta\sin\chi}{mV^2\cos\gamma}\operatorname{sgn}A_1$$
$$+\frac{\rho A}{2m}\frac{\sin\phi}{\cos\gamma}\left[C_L + \frac{V}{a}\frac{\partial C_L}{\partial M}\right]$$

$$\frac{\partial H}{\partial \gamma} = -\frac{V\sin\gamma\sin\psi\sin\lambda}{r\cos\lambda} + \frac{r\omega^2\sin\lambda\cos\lambda\sin\psi\sin\gamma}{V\cos^2\gamma} - \frac{2\omega\cos\lambda\cos\psi}{\cos^2\gamma}$$
$$+\frac{T\cos\eta\sin\chi\sin\gamma}{mV\cos^2\gamma}\operatorname{sgn}A_1 + \frac{L\sin\phi\sin\gamma}{mV\cos^2\gamma}$$

$$\frac{\partial H}{\partial \psi} = \frac{V\cos\gamma\cos\psi\sin\lambda}{r\cos\lambda} + \frac{r\omega^2\sin\lambda\cos\lambda\cos\psi}{V\cos\gamma} + \frac{2\omega\cos\lambda\sin\psi\sin\gamma}{\cos\gamma}$$

$$\frac{\partial H}{\partial r} = -\frac{V\cos\gamma\sin\psi\sin\lambda}{r^2\cos\lambda} + \frac{\omega^2\sin\lambda\cos\lambda\sin\psi}{V\cos\gamma} - \frac{A_e\cos\eta\sin\chi}{mV\cos\gamma}\,\mathrm{sgn}A_1\frac{dp}{dh}$$

$$+\frac{VA\sin\phi}{2m\cos\gamma}\left[C_L\frac{d\rho}{dh} - \rho\frac{V}{a^2}\frac{da}{dh}\frac{\partial C_L}{\partial M}\right]\cos\phi$$

$$\frac{\partial H}{\partial \lambda} = \frac{V\cos\gamma\sin\psi}{r\cos^2\lambda} + 2\omega\cos\lambda + \frac{r\omega^2\sin\psi}{V\cos\gamma}\left[\cos^2\lambda - \sin^2\lambda\right] + \frac{2\omega\sin\lambda\cos\psi\tan\gamma}{\cos\gamma}$$

$$\frac{\partial H}{\partial m} = -\frac{T\cos\eta\sin\chi}{m^2V\cos\gamma}\,\mathrm{sgn}A_1 - \frac{L}{m^2V}\frac{\sin\phi}{\cos\gamma}$$

$$\frac{\partial H}{\partial \eta} = -\frac{T\sin\eta\sin\chi}{mV\cos\gamma}\,\mathrm{sgn}A_1 + \frac{\rho VA}{2m\cos\gamma}\left[\frac{\partial C_L}{\partial\alpha}\sin\phi\cos\phi\cos\chi - C_L\frac{\sin^2\phi\cos\phi}{\sin\chi\cos^2\eta}\right]$$

$$\frac{\partial H}{\partial \chi} = \frac{T\cos\eta\cos\chi}{mV\cos\gamma}\,\mathrm{sgn}A_1 + \frac{\rho VA}{2m\cos\gamma}\left[\frac{\partial C_L}{\partial\alpha}\sin^2\phi + C_L\frac{\cos\chi\cos\eta\cos^3\phi}{\sin\eta}\right]$$

$$\frac{\partial I}{\partial V} = \sin\gamma \qquad\qquad \frac{\partial K}{\partial V} = \frac{\cos\gamma\sin\psi}{r\cos\lambda}$$

$$\frac{\partial I}{\partial \gamma} = V\cos\gamma \qquad\qquad \frac{\partial K}{\partial \gamma} = -\frac{V\sin\gamma\sin\psi}{r\cos\lambda}$$

$$\frac{\partial J}{\partial V} = \frac{\cos\gamma\cos\psi}{r} \qquad\qquad \frac{\partial K}{\partial \psi} = \frac{V\cos\gamma\cos\psi}{r\cos\lambda}$$

$$\frac{\partial J}{\partial \gamma} = -\frac{V\sin\gamma\cos\psi}{r} \qquad\qquad \frac{\partial K}{\partial r} = -\frac{V\cos\gamma\sin\psi}{r^2\cos\lambda}$$

$$\frac{\partial J}{\partial \psi} = -\frac{V\cos\gamma\sin\psi}{r} \qquad\qquad \frac{\partial K}{\partial \lambda} = \frac{V\cos\gamma\sin\psi\sin\lambda}{r\cos^2\lambda}$$

$$\frac{\partial J}{\partial r} = -\frac{V\cos\gamma\cos\psi}{r^2}$$

TRANSFORMATION TO INERTIAL FRAME

$$V_I = \sqrt{V^2 + 2(V\cos\gamma\sin\psi)(\omega r\cos\lambda) + (\omega r\cos\lambda)^2}$$

$$\sin\gamma_I = \frac{V\sin\gamma}{V_I}$$

$$\sin\psi_I = \frac{V\cos\gamma\sin\psi + \omega r\cos\lambda}{V_I\cos\gamma_I}$$

PARTIAL DERIVATIVES OF INERTIAL VARIABLES

$$\frac{\partial V_I}{\partial V} = \frac{V + (\omega r\cos\lambda)\cos\gamma\sin\psi}{(V_I)}$$

$$\frac{\partial V_I}{\partial r} = \frac{(V\cos\gamma\sin\psi)(\omega r\cos\lambda) + (\omega r\cos\lambda)^2}{r(V_I)}$$

$$\frac{\partial V_I}{\partial \gamma} = -(\omega r\cos\lambda)(\sin\gamma_I)\sin\psi$$

$$\frac{\partial V_I}{\partial \psi} = \frac{(\omega r\cos\lambda)V\cos\gamma\cos\psi}{V_I}$$

$$\frac{\partial V_I}{\partial \lambda} = -\frac{\sin\lambda}{\cos\lambda}\left[\frac{(V\cos\gamma\sin\psi)(\omega r\cos\lambda) + (\omega r\cos\lambda)^2}{V_I}\right]$$

$$\frac{\partial \gamma_I}{\partial V} = \left(\frac{\tan\gamma_I}{V_I}\right)\left[\frac{V_I}{V} - \left(\frac{\partial V_I}{\partial V}\right)\right]$$

$$\frac{\partial \gamma_I}{\partial r} = -\left(\frac{\tan\gamma_I}{V_I}\right)\left(\frac{\partial V_I}{\partial r}\right)$$

$$\frac{\partial \gamma_I}{\partial \gamma} = \frac{V}{V_I}\frac{\cos\gamma}{\cos\gamma_I} - \left(\frac{\tan\gamma_I}{V_I}\right)\left(\frac{\partial V_I}{\partial \gamma}\right)$$

$$\frac{\partial \gamma_I}{\partial \Psi} = -\left(\frac{\tan \gamma_I}{V_I}\right)\left(\frac{\partial V_I}{\partial \Psi}\right)$$

$$\frac{\partial \gamma_I}{\partial \lambda} = -\left(\frac{\tan \gamma_I}{V_I}\right)\left(\frac{\partial V_I}{\partial \lambda}\right)$$

$$\frac{\partial \Psi_I}{\partial V} = \frac{-(\omega r \cos \lambda)}{V}\left(\frac{\cos^2 \Psi_I}{V \cos \gamma \cos \Psi}\right)$$

$$\frac{\partial \Psi_I}{\partial r} = \frac{(\omega r \cos \lambda)}{r}\left(\frac{\cos^2 \Psi_I}{V \cos \gamma \cos \Psi}\right)$$

$$\frac{\partial \Psi_I}{\partial \gamma} = (\omega r \cos \lambda)(\tan \gamma)\left(\frac{\cos^2 \Psi_I}{V \cos \gamma \cos \Psi}\right)$$

$$\frac{\partial \Psi_I}{\partial \Psi} = \frac{\cos^2 \Psi_I}{\cos^2 \Psi} + \frac{\sin \Psi}{\cos \Psi}(\omega r \cos \lambda)\left(\frac{\cos^2 \Psi_I}{V \cos \gamma \cos \Psi}\right)$$

$$\frac{\partial \Psi_I}{\partial \lambda} = -\omega r \sin \lambda\left(\frac{\cos^2 \Psi_I}{V \cos \gamma \cos \Psi}\right)$$

TERMINAL CONSTRAINT EQUATIONS

The nature of the trajectory optimization process in PRESTO is virtually the same over all the space missions available. The differences lie only in the form of the terminal constraints imposed on the trajectory. For the Earth-orbit mission terminal constraints can be imposed on the trajectory variables explicitly, or they can be specified in terms of conventional orbit elements which are functions of the trajectory variables. For the lunar and planetary transfer missions the terminal constraints have been formulated as parameters which are functions of Earth orbit elements. Thus, multiple use of the coding is made possible and the form of the constraints is conceptually similar among the three types of mission. The various constraint parameters are discussed, by mission, and defined in equation form in this section. In addition, the partial derivatives of the constraint parameters with respect to the inertial trajectory variables are documented. These derivatives are used in setting initial conditions on the adjoint variables to be integrated.

Earth-Orbit Mission

The orbit elements are diagrammed in the figure on the following page. The equator constitutes the basic reference plane. The inertial "longitude" reference is in the direction of the vernal equinox, which is defined here to be the intersection of the plane of the ecliptic 1950.0 and the equator of date.

The form of the equations used for the orbit elements is such that they are valid for elliptic, parabolic and hyperbolic orbits.

Lunar Transfer Mission

In this mission, one is interested in boosting to burnout conditions that will lead to lunar intercept in a stated length of time. In PRESTO conic motion is assumed for the transfer arc, and numerical integration of the trajectory is conducted only to burnout. Burnout, on any iteration, is the time at which the (instant) combination of trajectory variables would produce a coast to lunar radial distance in the specified transfer time. Thus, transfer time to lunar radial distance constitutes the stopping parameter. The remaining requirement is that the vehicle position at the end of this coast be the same as the Moon (intercept). This position (angles of right ascension and declination) is computed from the trajectory variables at burnout, again assuming conic transfer to the Moon. These two angles constitute the terminal constraint parameters; their <u>desired</u> values are found automatically by the ephemeris routine.

It should be realized that with the above formulation the optimum injection conditions are found as an implicit part of the trajectory shaping process.

Finally, transfer orbit inclination can also be constrained to a specified value by input option.

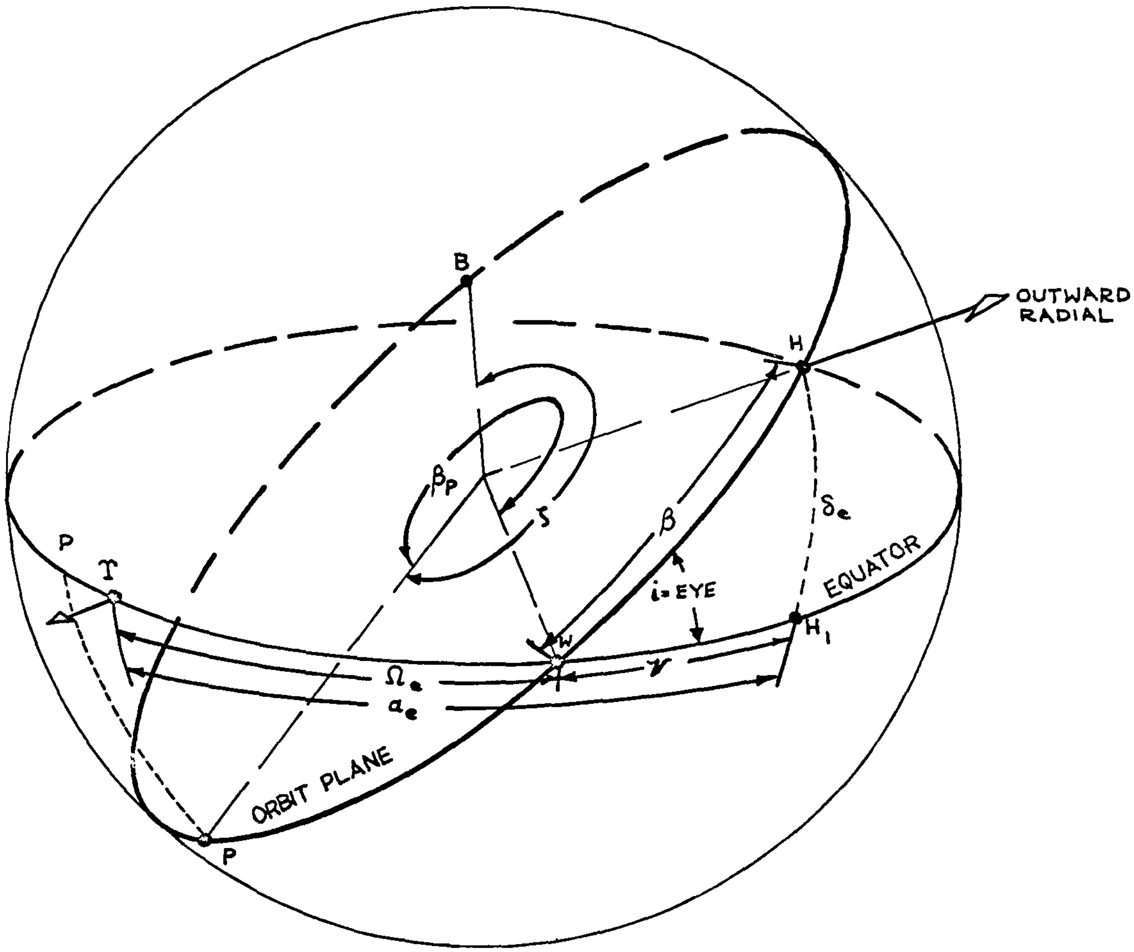
Planetary Transfer Mission

The terminal constraints formulation for the planetary transfer mission is very analogous to that for the lunar mission. In this case, the <u>magnitude</u>

of the hyperbolic excess velocity vector constitutes the stopping parameter, and its right ascension and declination direction angles are the constraint parameters. (The direction of the vector is that of the asymptote to the departure hyperbola.) As discussed in Section 9, the characteristics of the departure hyperbola required for a specified transfer are computed by the PLANEP subroutine. (They can also be input directly.) Further, an approximation is allowed as follows. Since the direction of the asymptote is stated in terms of its right ascension and declination, an Earth-centered origin of the asymptote is implied. The approximation employed is to accept any hyperbola (of the required energy) whose asymptote is parallel to the required direction. Since typically the asymptote passes within one or two Earth-radii of Earth center, the approximation is relatively slight.

Again, it should be realized that with the above formulation the optimum injection conditions are found as an implicit part of the trajectory optimization.

GEOMETRY USED TO DEFINE THE ORBIT PLANE FEATURES
IN SPACE--ORIENTED EQUATORIAL SYSTEM



OUTWARD RADIAL

- For Planetary Mission--Direction of hyperbolic excess velocity vector translated to Earth center
- For Lunar Mission--Vehicle position at Lunar radial distance

B	Booster burnout	α_e	Right ascension of Outward Radial
H	Direction of Outward Radial	β	In-plane angle from ascending node
H_1	Equatorial projection of H	β_p	Argument of perigee
P	Perigee of orbit	i = EYE	Inclination of geocentric orbit plane
P_1	Equatorial projection of P	δ_e	Declination of Outward Radial
Υ	Vernal equinox	ν	Inertial longitude angle from ascending node
ζ	True anomaly	Ω_e	Longitude of W
		W	Ascending node of the orbit plane

ORBIT ELEMENTS

ENERGY (x2)

$$2E = V_I^2 - \frac{2\mu}{r}$$

ANGULAR MOMENTUM

$$H = r V_I \cos \gamma_I$$

PERIGEE RADIUS

$$r_p = \frac{H}{\left(\frac{\mu}{H}\right) + \sqrt{2E + \left(\frac{\mu}{H}\right)^2}}$$

INCLINATION

$$i = \cos^{-1}(\cos\lambda \sin\psi_I) \qquad 0 \le i \le \pi$$

$$\text{IF } (\sin\psi_I) \quad \begin{matrix} \left.\begin{matrix} 0 \\ + \end{matrix}\right\} 0 \le i \le \frac{\pi}{2} \\ -\} \frac{\pi}{2} < i \le \pi \end{matrix}$$

LONGITUDE OF ASCENDING NODE

$$\Omega_e = \alpha_e - \nu$$

WHERE α_e = right ascension of vehicle position
ν = inertial longitude angle from ascending node
$\alpha_e = 99.°659967 + 0.°98564743 T_G + \tau^\circ + 360^\circ(T_G - [T_G])$
τ = Longitude, degrees east
$(T_G - [T_G])$ = Greenwich time of day.

$$0^\circ \le \alpha_e < 360^\circ$$

$$\nu = \tan^{-1}\left(\frac{\sin\psi_I \sin\lambda}{\cos\psi_I}\right) \qquad 0 \le \nu < 2\pi$$

use 4 quadrant $\tan^{-1}$

$$0 \le \Omega_e < 360^\circ$$

ARGUMENT OF PERIGEE

$$\beta_p = (\beta - \mathcal{J})$$

β = in-plane range angle from ascending node
$\mathcal{J}$ = true anomaly

WHERE $\beta = \tan^{-1}\left(\frac{\sin\lambda}{\cos\lambda\cos\Psi_I}\right)$ $\qquad 0 \le \beta < 2\pi$

use 4 quadrant $\tan^{-1}$

& $\mathcal{J} = \tan^{-1}\left[\frac{\tan\gamma_I}{1-\left(\frac{\mu}{H}\right)\left(\frac{r}{H}\right)}\right]$ $\qquad 0 \le \mathcal{J} < 2\pi$

use 4 quadrant $\tan^{-1}$

& $0 \le \beta_p < 360°$

TEST IF $(2E)$ $\left.\begin{matrix}0\\+\end{matrix}\right\}$ EXIT

$-\}$ PROCEED

SEMI MAJOR AXIS

$$r_\alpha = \frac{-\mu}{2E}$$

TIME FROM PERICENTER

$$t_{\mathcal{J}} = r_\alpha\left[\frac{1}{\sqrt{-2E}}\cdot Z - \frac{\tan\gamma_I}{\left(\frac{\mu}{H}\right)}\right]$$

WHERE $Z = \cos^{-1}\left(\frac{r_\alpha - r}{r_\alpha - r_p}\right)$ $\qquad 0 \le Z \le \pi$

AND IF (γ_I) $\left.\begin{matrix}0\\+\end{matrix}\right\}$ $Z = Z$

$-\}$ $Z = 2\pi - Z$

DERIVATIVES OF ORBIT ELEMENTS

ENERGY

$$\frac{\partial E}{\partial V_I} = V_I \qquad \frac{\partial E}{\partial r} = \frac{\mu}{r^2}$$

ANGULAR MOMENTUM

$$\frac{\partial H}{\partial V_I} = r \cos \gamma_I$$

$$\frac{\partial H}{\partial r} = V_I \cos \gamma_I$$

$$\frac{\partial H}{\partial \gamma_I} = -r V_I \sin \gamma_I$$

PERIGEE RADIUS

$$\frac{\partial r_p}{\partial V_I} = \frac{\left(\frac{\partial H}{\partial V_I}\right) - \left(\frac{r_p^2}{H}\right)\left(\frac{\partial E}{\partial V_I}\right)}{\sqrt{2E + \left(\frac{\mu}{H}\right)^2}}$$

$$\frac{\partial r_p}{\partial r} = \frac{\left(\frac{\partial H}{\partial r}\right) - \left(\frac{r_p^2}{H}\right)\left(\frac{\partial E}{\partial r}\right)}{\sqrt{2E + \left(\frac{\mu}{H}\right)^2}}$$

$$\frac{\partial r_p}{\partial \gamma_I} = \frac{\left(\frac{\partial H}{\partial \gamma_I}\right)}{\sqrt{2E + \left(\frac{\mu}{H}\right)^2}}$$

INCLINATION

$$\frac{\partial i}{\partial \lambda} = \frac{\sin\lambda \sin\Psi_I}{\sin i}$$

$$\frac{\partial i}{\partial \Psi_I} = -\frac{\cos\lambda \cos\Psi_I}{\sin i}$$

LONGITUDE OF ASCENDING NODE

$$\frac{\partial \Omega e}{\partial \lambda} = -\frac{\cos i \cos\Psi_I}{\sin^2 i}$$

$$\frac{\partial \Omega e}{\partial \Psi_I} = -\frac{\sin\lambda}{\sin^2 i}$$

$$\frac{\partial \Omega e}{\partial \tau} = 1$$

$$\frac{\partial \Omega e}{\partial T_G} = 4.1068643 \times 10^{-3} \text{ deg/sec.}$$

ARGUMENT OF PERIGEE

$$\frac{\partial \beta_p}{\partial V_I} = \frac{2(\sin^2 \mathcal{J})\left(\frac{\mu}{H}\right)\left(\frac{r}{H}\right)}{V_I \tan\gamma_I}$$

$$\frac{\partial \beta_p}{\partial r} = \frac{\sin^2 \mathcal{J}\left(\frac{\mu}{H}\right)}{H \tan\gamma_I}$$

$$\frac{\partial \beta_p}{\partial \gamma} = -(\sin^2 \mathcal{J})\left[\frac{1-\left(\frac{\mu}{H}\right)\left(\frac{r}{H}\right)}{\sin^2\gamma_I} + 2\left(\frac{\mu}{H}\right)\left(\frac{r}{H}\right)\right]$$

$$\frac{\partial \beta_p}{\partial \lambda} = \frac{\sin\beta \cos\beta}{\sin\lambda \cos\lambda}$$

$$\frac{\partial \beta_p}{\partial \psi_I} = \frac{\sin\beta \cos\beta \sin\psi_I}{\cos\psi_I}$$

TIME FROM PERICENTER

$$\frac{\partial t_f}{\partial V_I} = -\frac{t_f V_I}{E} + \frac{r_\alpha}{\sqrt{-2E}}\left(\frac{\partial Z}{\partial V_I} - \frac{Z V_I}{2E}\right) - \frac{r_\alpha \tan\gamma_I}{\left(\frac{\mu}{H}\right) V_I}$$

WHERE $$\frac{\partial Z}{\partial V_I} = \frac{r_\alpha \frac{V_I}{E} - (\cos Z)\left(\frac{r_\alpha V_I}{E} + \frac{\partial r_p}{\partial V_I}\right)}{(r_\alpha - r_p)\sqrt{1-\cos^2 Z}}$$

$$\frac{\partial t_I}{\partial r} = \frac{r_\alpha}{r}\left(\frac{2t_I}{r} - \frac{\tan\gamma_I}{\left(\frac{\mu}{H}\right)}\right) + \frac{r_\alpha}{\sqrt{-2E}}\left(\frac{\partial Z}{\partial r} - \frac{Z}{2E}\frac{\mu}{r^2}\right)$$

WHERE $$\frac{\partial Z}{\partial r} = \frac{1 - 2\left(\frac{r_\alpha}{r}\right)^2(1-\cos Z) - \cos Z \frac{\partial r_p}{\partial r}}{(r_\alpha - r_p)\sqrt{1-\cos^2 Z}}$$

$$\frac{\partial t_f}{\partial \gamma_I} = r_\alpha \left[\frac{\left(\frac{\partial Z}{\partial \gamma_I}\right)}{\sqrt{-2E}} - \frac{1}{\left(\frac{\mu}{H}\right)}\right]$$

WHERE $$\frac{\partial Z}{\partial \gamma_I} = \frac{-\cos Z\left(\frac{\partial r_p}{\partial \gamma_I}\right)}{(r_\alpha - r_p)\left(\sqrt{1-\cos^2 Z}\right)}$$

LUNAR TRANSFER TERMINAL CONSTRAINTS

1. Use as a stopping parameter the transfer time to lunar radial distance -- time being measured from perigee of transfer ellipse.

$$t_M = \frac{\sqrt{2}}{5400}\,\frac{r_P}{E}\,\frac{\left(2+\frac{r_M}{r_P}\right)}{\sqrt{\frac{r_M}{r_P}-1}}\left(\sqrt{E\left(\frac{r_M}{r_P}+1\right)+\frac{\mu}{r_P}}-\sqrt{2E+\frac{\mu}{r_P}}\right)$$

r_P = perigee radius

r_M = lunar radial distance as determined from ephemeris routine for nominal arrival time

E = transfer orbit energy

2. As the first of two constraints use the computed declination of the vehicle at r_M

$$\delta_{eM} = \sin^{-1}(\sin i \sin \beta_M) \qquad -\frac{\pi}{2} \le \delta_e \le \frac{\pi}{2}$$

i = inclination (either constrained or just evaluated)

$\beta_M = \beta_P + \mathcal{L}_M$

β_P = argument of perigee

$\mathcal{L}_M$ = true anomaly @ r_M

$$\mathcal{L}_M = \tan^{-1}\left[\frac{\sqrt{\left(2E+\frac{2\mu}{r_M}\right)\left(\frac{r_M}{H}\right)^2-1}}{1-\left(\frac{\mu}{H}\right)\left(\frac{r_M}{H}\right)}\right]$$ use 4 quad $\tan^{-1}$
use H from above

3. As the second contraints, use the computed right ascension of the vehicle at r_M.

$\alpha e_M = \Omega e + \nu_M$

Ωe = Longitude of Ascending Node

$\nu_M = \tan^{-1}\left(\frac{\cos i \sin \beta_M}{\cos \beta_M}\right)$ use 4 quad $\tan^{-1}$ $0 \le \nu < 2\pi$

4. In addition, the inclination (i) may (or may not) be constrained via input option.

5. t_M is always input

$\left.\begin{array}{l} i \\ r_M \\ \delta e_M \\ \alpha e_M \end{array}\right\}$ may be input

If use of ephemeris is selected, $r_M, \delta e_M, \alpha e_M$ are obtained from it in the following manner:

An initial estimate of time of arrival is made from

$T_{G_{M_0}} = T_{G_0} + \frac{t_M}{24}$ (T_{G_0} = launch time)

and the lunar ephemeris is entered with this time to obtain $r_M, \delta e_0, \alpha e_0$. Then the ephemeris is entered again, using a time 1/2 day later, to determine the Rates of change of δe_M and αe_M.

Thus, @ $T_{GM_1} = T_{GM_0} + 0.5$ obtain δe_1 and αe_1, then for each trajectory iteration, compute the desired δe_M and αe_M from

$$\delta e_M = \delta e_0 + \frac{(T_{GM} - T_{GM_0})}{0.5}(\delta e_1 - \delta e_0)$$

$$\alpha e_M = \alpha e_0 + \frac{(T_{GM} - T_{GM_0})}{0.5}(\alpha e_1 - \alpha e_0)$$

where T_{GM} is computed from the previous iteration from:

$$T_{GM} = T_{G_{@\,\text{terminal condition}}} + (t_M - t_J)\left(\frac{1}{24}\right)$$

t_J = time from perigee computed @ terminal condition

PARTIAL DERIVATIVES OF LUNAR TRANSFER CONSTRAINTS

1. TRANSFER TIME (nominal stopping parameter)

$$\frac{\partial t_M}{\partial V_I} = \frac{\partial t_M}{\partial E}\frac{\partial E}{\partial V_I} + \frac{\partial t_M}{\partial r_p}\frac{\partial r_p}{\partial V_I}$$

$$\frac{\partial t_M}{\partial r} = \frac{\partial t_M}{\partial E}\frac{\partial E}{\partial r} + \frac{\partial t_M}{\partial r_p}\frac{\partial r_p}{\partial r}$$

$$\frac{\partial t_M}{\partial \gamma_I} = \frac{\partial t_M}{\partial r_p}\frac{\partial r_p}{\partial \gamma_I}$$

where $$\frac{\partial t_M}{\partial E} = \frac{\sqrt{2}}{5400}\frac{r_M}{E^2}\frac{\left(1+2\frac{r_p}{r_M}\right)}{\sqrt{1-\frac{r_p}{r_M}}}\left[\frac{E\frac{r_p}{r_M}+\frac{\mu}{r_M}}{\sqrt{2E\frac{r_p}{r_M}+\frac{\mu}{r_M}}} - \frac{E\left(1+\frac{r_p}{r_M}\right)+2\frac{\mu}{r_M}}{2\sqrt{E\left(1+\frac{r_p}{r_M}\right)+\frac{\mu}{r_M}}}\right]$$

$$\frac{\partial t_M}{\partial r_p} = \frac{\sqrt{2}}{10{,}800\,E\left(1-\frac{r_p}{r_M}\right)^{3/2}}\left\{\frac{2E\left[3+2\frac{r_p}{r_M}-2\left(\frac{r_p}{r_M}\right)^2\right]+\frac{\mu}{r_M}\left(5-2\frac{r_p}{r_M}\right)}{\sqrt{E\left(1+\frac{r_p}{r_M}\right)+\frac{\mu}{r_M}}}\right.$$

$$\left. - \frac{2E\left[1+6\frac{r_p}{r_M}-4\left(\frac{r_p}{r_M}\right)^2\right]+\frac{\mu}{r_M}\left(5-2\frac{r_p}{r_M}\right)}{\sqrt{2E\frac{r_p}{r_M}+\frac{\mu}{r_M}}}\right\}$$

2. DECLINATION

$$\frac{\partial \delta_{eM}}{\partial V_I} = \frac{\sin i \cos \beta_M}{\cos \delta_{eM}} \left(\frac{\partial \beta_M}{\partial V_I} \right)$$

$$\frac{\partial \beta_M}{\partial V_I} = \frac{\partial \beta_P}{\partial V_I} + \frac{\partial \mathcal{J}_M}{\partial V_I}$$

$$\frac{\partial \mathcal{J}_M}{\partial V_I} = \frac{\tan \mathcal{J}_M}{V_I \left(1+\tan^2 \mathcal{J}_M\right) \sqrt{\left(2E+\frac{2\mu}{r_M}\right)\left(\frac{r_M}{H}\right)^2 - 1}} \left\{ \frac{V_I^2 \left(\frac{r_M}{H}\right)^2 - 1}{\sqrt{\left(2E+\frac{2\mu}{r_M}\right)\left(\frac{r_m}{H}\right)^2 - 1}} - \tan \mathcal{J}_M \left[1 + \left(\frac{\mu}{H}\right)\left(\frac{r_m}{H}\right) \right] \right\}$$

$$\frac{\partial \delta_{eM}}{\partial r} = \frac{\sin i \cos \beta_M}{\cos \delta_{eM}} \left(\frac{\partial \beta_M}{\partial r} \right)$$

$$\frac{\partial \beta_M}{\partial r} = \frac{\partial \beta_P}{\partial r} + \frac{\partial \mathcal{J}_M}{\partial r}$$

$$\frac{\partial \mathcal{J}_M}{\partial r} = \frac{\tan \mathcal{J}_M}{r \left(1+\tan^2 \mathcal{J}_M\right) \sqrt{\left(2E+\frac{2\mu}{r_M}\right)\left(\frac{r_M}{H}\right)^2 - 1}} \left\{ \frac{\left(\frac{\mu}{H}\right)\left(\frac{r_M}{H}\right)\left(\frac{r_M}{r}\right) - 1}{\sqrt{\left(2E+\frac{2\mu}{r_M}\right)\left(\frac{r_M}{H}\right)^2 - 1}} - \tan \mathcal{J}_M \left[1 + \left(\frac{\mu}{H}\right)\left(\frac{r_M}{H}\right) \right] \right\}$$

$$\frac{\partial \delta_{eM}}{\partial \gamma_I} = \frac{\sin i \cos \beta_M}{\cos \delta_{eM}} \left(\frac{\partial \beta_M}{\partial \gamma_I} \right)$$

$$\frac{\partial \beta_M}{\partial \gamma_I} = \frac{\partial \beta_P}{\partial \gamma_I} + \frac{\partial \mathcal{J}_M}{\partial \gamma_I}$$

$$\frac{\partial \mathcal{J}_M}{\partial \gamma_I} = \frac{\tan \gamma_I \tan \mathcal{J}_M}{\left(1+\tan^2 \mathcal{J}_M\right) \sqrt{\left(2E+\frac{2\mu}{r_M}\right)\left(\frac{r_M}{H}\right)^2 - 1}} \left\{ \frac{1}{\sqrt{\left(2E+\frac{2\mu}{r_M}\right)\left(\frac{r_M}{H}\right)^2 - 1}} + \tan \mathcal{J}_M \left[1 + \left(\frac{\mu}{H}\right)\left(\frac{r_M}{H}\right) \right] \right\}$$

$$\frac{\partial \delta_{eM}}{\partial \lambda} = \frac{1}{\cos \delta_{eM}} \left[\sin i \cos \beta_M \frac{\partial \beta_M}{\partial \lambda} + \sin \beta_M \cos i \frac{\partial i}{\partial \lambda} \right]$$

$$\frac{\partial \beta_M}{\partial \lambda} = \frac{\partial \beta_P}{\partial \lambda}$$

$$\frac{\partial \delta_{e_M}}{\partial \Psi_I} = \frac{1}{\cos \delta_{e_M}}\left[\sin i \cos \beta_M \frac{\partial \beta_M}{\partial \Psi_I} + \sin \beta_M \cos i \frac{\partial i}{\partial \Psi_I}\right]$$

$$\frac{\partial \beta_M}{\partial \Psi_I} = \frac{\partial \beta_P}{\partial \Psi_I}$$

3. RIGHT ASCENSION

$$\frac{\partial \alpha_{e_M}}{\partial V_I} = \frac{\partial \nu_M}{\partial V_I} = \frac{1}{(1+\tan^2 \nu_M)}\left(\frac{\cos i}{\cos^2 \beta_M}\right)\left(\frac{\partial \beta_M}{\partial V_I}\right)$$

$$\frac{\partial \alpha_{e_M}}{\partial r} = \frac{\partial \nu_M}{\partial r} = \frac{1}{(1+\tan^2 \nu_M)}\left(\frac{\cos i}{\cos^2 \beta_M}\right)\left(\frac{\partial \beta_M}{\partial r}\right)$$

$$\frac{\partial \alpha_{e_M}}{\partial \gamma_I} = \frac{\partial \nu_M}{\partial \gamma_I} = \frac{1}{(1+\tan^2 \nu_M)}\left(\frac{\cos i}{\cos^2 \beta_M}\right)\left(\frac{\partial \beta_M}{\partial \gamma_I}\right)$$

$$\frac{\partial \alpha_{e_M}}{\partial \lambda} = \frac{\partial \Omega_e}{\partial \lambda} + \frac{\partial \nu_M}{\partial \lambda} = \frac{\partial \Omega_e}{\partial \lambda} + \frac{1}{(1+\tan^2 \nu_M)}\left[\frac{\cos i}{\cos^2 \beta_M}\frac{\partial \beta_M}{\partial \lambda} - \tan \beta_M \sin i \frac{\partial i}{\partial \lambda}\right]$$

$$\frac{\partial \alpha_{e_M}}{\partial \Psi_I} = \frac{\partial \Omega_e}{\partial \Psi_I} + \frac{\partial \nu_M}{\partial \Psi_I} = \frac{\partial \Omega_e}{\partial \Psi_I} + \frac{1}{(1+\tan^2 \nu_M)}\left[\frac{\cos i}{\cos^2 \beta_M}\frac{\partial \beta_M}{\partial \Psi_I} - \tan \beta_M \sin i \frac{\partial i}{\partial \Psi_I}\right]$$

$$\frac{\partial \alpha_{e_M}}{\partial \tau} = \frac{\partial \Omega_e}{\partial \tau}$$

$$\frac{\partial \alpha_{e_M}}{\partial T_G} = \frac{\partial \Omega_e}{\partial T_G}$$

PLANETARY TRANSFER TERMINAL CONSTRAINTS

1. Use as a stopping parameter the hyperbolic excess velocity (V_H).

$$V_H = \sqrt{V_I^2 - \frac{2\mu}{r}} = \sqrt{2E}$$

2. Use as the first of two constraints the declination of the hyperbolic asymptote (δe_H)

$$\delta e_H = \sin^{-1}(\sin i \sin(\beta_H)) \quad -\frac{\pi}{2} \leq \delta e_H \leq \frac{\pi}{2}$$

where: i = inclination $\quad \beta_p$ = argument of perigee

$\beta_H = \beta_p + \mathcal{J}_H \qquad \mathcal{J}_H$ = limiting value of true anomaly of hyperbola

$\mathcal{J}_H = \pi - \tan^{-1}\frac{HV_H}{\mu} \qquad H$ = Angular momentum

3. The second constraint will be the right ascension of the hyperbolic asymptote (αe_H)

$\alpha e_H = \Omega e + \nu_H \quad \Omega e$ = Longitude of Ascending Node

$$\nu_H = \tan^{-1}\left(\frac{\cos i \sin \beta_H}{\cos \beta_H}\right) \quad 0 \leq \nu < 2\pi \text{ use 4 quad } \tan^{-1}$$

4. Values of V_H, δe_H, αe_H will be obtained from input or from Stan Ross program as a subroutine.

$$\frac{\partial \beta_H}{\partial \gamma_I} = \frac{\partial \beta_P}{\partial \gamma_I} + \frac{\partial \mathcal{J}_H}{\partial \gamma_I}$$

$$\frac{\partial \mathcal{J}_H}{\partial \gamma_I} = \frac{\left(\frac{H V_H}{\mu}\right)\tan \gamma_I}{1 + \left(\frac{H V_H}{\mu}\right)^2}$$

$$\frac{\partial \delta e_H}{\partial \lambda} = \frac{1}{\sqrt{1-\sin^2 \delta e_H}}\left(\sin i \cos \beta_H \frac{\partial \beta_H}{\partial \lambda} + \sin \beta_H \cos i \frac{\partial i}{\partial \lambda}\right)$$

$$\frac{\partial \beta_H}{\partial \lambda} = \frac{\partial \beta_P}{\partial \lambda}$$

$$\frac{\partial \delta e_H}{\partial \psi_I} = \frac{1}{\sqrt{1-\sin^2 \delta e_H}}\left(\sin i \cos \beta_H \frac{\partial \beta_H}{\partial \psi_I} + \sin \beta_H \cos i \frac{\partial i}{\partial \psi_I}\right)$$

$$\frac{\partial \beta_H}{\partial \psi_I} = \frac{\partial \beta_P}{\partial \psi_I}$$

3. RIGHT ASCENSION

$$\frac{\partial \alpha e_H}{\partial V_I} = \left(\frac{\cos i / \cos^2 \beta_H}{1 + \tan^2 \nu_H}\right)\left(\frac{\partial \beta_H}{\partial V_I}\right)$$

$$\frac{\partial \alpha e_H}{\partial r} = \left(\frac{\cos i / \cos^2 \beta_H}{1 + \tan^2 \nu_H}\right)\left(\frac{\partial \beta_H}{\partial r}\right) + \left(\frac{\partial \Omega_e}{\partial \psi_I}\right)\left(\frac{\partial \psi_I}{\partial r}\right)$$

$$\frac{\partial \alpha e_H}{\partial \gamma_I} = \left(\frac{\cos i / \cos^2 \beta_H}{1 + \tan^2 \nu_H}\right)\left(\frac{\partial \beta_H}{\partial \gamma_I}\right)$$

$$\frac{\partial \alpha e_H}{\partial \lambda} = \frac{\partial \Omega e}{\partial \lambda} + \left(\frac{\cos i / \cos^2 \beta_H}{1 + \tan^2 \nu_H}\right)\left(\frac{\partial \beta_H}{\partial \lambda}\right) - \frac{\tan \beta_H}{1 + \tan^2 \nu_H} \sin i \left(\frac{\partial i}{\partial \lambda}\right)$$

$$\frac{\partial \alpha e_H}{\partial \psi_I} = \frac{\partial \Omega e}{\partial \psi_I} + \left(\frac{\cos i / \cos^2 \beta_H}{1 + \tan^2 \nu_H}\right)\left(\frac{\partial \beta_H}{\partial \psi_I}\right) - \frac{\tan \beta_H}{1 + \tan^2 \nu_H} \sin i \left(\frac{\partial i}{\partial \psi_I}\right)$$

$$\frac{\partial \alpha e_H}{\partial \tau} = \frac{\partial \Omega e}{\partial \tau} \qquad \frac{\partial \alpha e_H}{\partial T_G} = \frac{\partial \Omega e}{\partial T_G}$$

PARTIAL DERIVATIVES OF PLANETARY TRANSFER CONSTRAINTS

1. HYPERBOLIC EXCESS VELOCITY (Nominal stopping parameter)

$$\frac{\partial V_H}{\partial V_I} = \frac{2 V_I}{2 V_H} = \frac{V_I}{V_H}$$

$$\frac{\partial V_H}{\partial r} = \frac{2 \frac{\mu}{r^2}}{2 V_H} = \frac{\mu / r^2}{V_H}$$

2. DECLINATION

$$\frac{\partial \delta e_H}{\partial V_I} = \frac{\sin i \cos \beta_H}{\sqrt{1 - \sin^2 \delta e_H}} \frac{\partial \beta_H}{\partial V_I}$$

$$\frac{\partial \beta_H}{\partial V_I} = \frac{\partial \beta_P}{\partial V_I} + \frac{\partial \int_H}{\partial V_I}$$

$$\frac{\partial \int_H}{\partial V_I} = -\frac{\left(\frac{H V_H}{\mu}\right)\left[1 + \left(\frac{V_I}{V_H}\right)^2\right]}{V_I \left[1 + \left(\frac{H V_H}{\mu}\right)^2\right]}$$

$$\frac{\partial \delta e_H}{\partial r} = \frac{\sin i \cos \beta_H}{\sqrt{1 - \sin^2 \delta e_H}} \frac{\partial \beta_H}{\partial r}$$

$$\frac{\partial \beta_H}{\partial r} = \frac{\partial \beta_P}{\partial r} + \frac{\partial \int_H}{\partial r}$$

$$\frac{\partial \int_H}{\partial r} = -\frac{\left(\frac{H V_H}{\mu}\right)\left[1 + \frac{\mu}{r V_H^2}\right]}{r\left[1 + \left(\frac{H V_H}{\mu}\right)^2\right]}$$

$$\frac{\partial \delta e_H}{\partial \gamma_I} = \frac{\sin i \cos \beta_H}{\sqrt{1 - \sin^2 \delta e_H}} \frac{\partial \beta_H}{\partial \gamma_I}$$

Coast Equations

Given state variables at start of coast and specified range angle of coast (β_c), evaluate state variables at end of coast, including time duration of coast.

1. Compute: (1) $2E = V_{I_1}^2 - \frac{2\mu}{r_1}$

(2) $H = r_1 V_{I_1} \cos \gamma_{I_1}$

(3) $r_p = \frac{H}{\left(\frac{\mu}{H}\right) + \sqrt{2E + \left(\frac{\mu}{H}\right)^2}}$

(4) $i = \cos^{-1}(\cos \lambda_1 \sin \psi_{I_1}) \qquad 0 \le i \le \pi$

(5) $\nu_1 = \tan^{-1}\left(\frac{\sin \psi_{I_1} \sin \lambda_1}{\cos \psi_{I_1}}\right) \qquad 0 \le \nu < 2\pi$

(6) $\beta_1 = \tan^{-1}\left(\frac{\sin \lambda_1}{\cos \lambda_1 \cos \psi_{I_1}}\right) \qquad 0 \le \beta < 2\pi$

(7) $\mathcal{J}_1 = \tan^{-1}\left[\frac{\tan \gamma_{I_1}}{1 - \left(\frac{\mu}{H}\right)\left(\frac{r}{H}\right)}\right] \qquad 0 \le \mathcal{J} < 2\pi$

TEST: IF $(2E)$ $\left.\begin{matrix} 0 \\ + \end{matrix}\right\}$ skip next two computations

$-\}$ proceed

(8) $r_\alpha = -\frac{\mu}{2E}$

(9) $t_{\mathcal{J}_1} = r_\alpha \left[\frac{1}{\sqrt{-2E}} \cdot Z - \frac{\tan \gamma_{I_1}}{\left(\frac{\mu}{H}\right)}\right]$

2. Increment: $\mathcal{J}_2 = \mathcal{J}_1 + \beta_c$ & $\beta_2 = \beta_1 + \beta_c$

3. COMPUTE: (1) $r_2 = \dfrac{H}{\left(\frac{\mu}{H}\right) + \sqrt{2E + \left(\frac{\mu}{H}\right)^2}\cos \mathcal{L}_2}$

(2) $V_{I_2} = \sqrt{2E + \frac{2\mu}{r_2}}$

(3) $\gamma_{I_2} = \cos^{-1}\left(\dfrac{H}{V_{I_2} r_2}\right)$ $\qquad -\frac{\pi}{2} \leq \gamma_I \leq \frac{\pi}{2}$

IF $(180 - \mathcal{L}_2)$ $\quad$ $0 \quad \gamma = 0$

$+ \quad \gamma = +|\gamma|$

$- \quad \gamma = -|\gamma|$

(4) $\lambda_2 = \sin^{-1}(\sin \beta_2 \sin i)$ $\quad -\frac{\pi}{2} \leq \lambda \leq \frac{\pi}{2}$

(standard 2 quadrant resolution)

(5) $\psi_{I_2} = \tan^{-1}\left(\dfrac{\cos i}{\cos \beta_2 \sin i}\right)$ $\qquad 0 \leq \psi < 2\pi$

use 4 quadrant $\tan^{-1}$

(6) $\nu_2 = \tan^{-1}\left(\dfrac{\sin \psi_{I_2} \sin \lambda_2}{\cos \psi_{I_2}}\right)$ $\qquad 0 \leq \nu < 2\pi$

use 4 quandrant $\tan^{-1}$

(7) $t_{\mathcal{J}_2} = r_\alpha\left[\dfrac{1}{\sqrt{-2E}} \cdot Z_2 - \dfrac{\tan \gamma_{I_2}}{\left(\frac{\mu}{H}\right)}\right]$

WHERE $Z_2 = \cos^{-1}\left(\dfrac{r_\alpha - r_2}{r_\alpha - r_p}\right)$ $\qquad 0 \leq Z < 2\pi$

AND IF (γ_{I_2}) $\quad \left.\begin{matrix} 0 \\ + \end{matrix}\right\} Z = Z$

$\left.-\right\} Z = 2\pi - Z$

(8) $t_2 = t_1 + (t_{\mathcal{J}_2} - t_{\mathcal{J}_1})$

(9) $\tau_2 = \tau_1 + (\nu_2 - \nu_1) - \omega(t_{\mathcal{J}_2} - t_{\mathcal{J}_1})$

DERIVATION AND PROGRAMMING OF OPTIMIZATION EQUATIONS

DERIVATION OF PROPERTIES OF ADJOINT VARIABLES

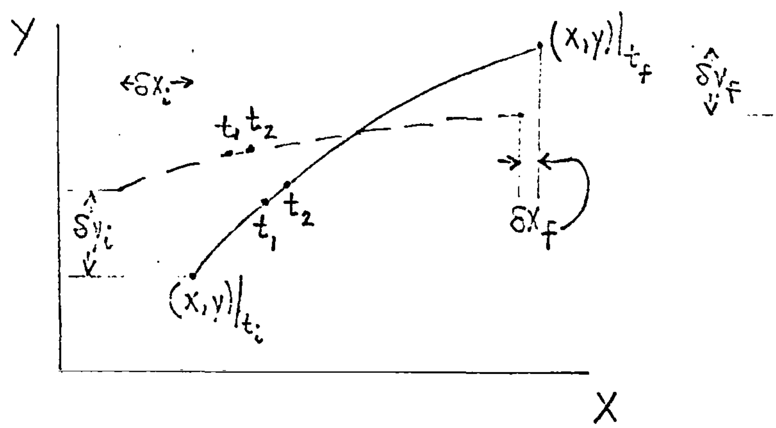
The steepest descent method of trajectory optimization depends on obtaining the effects of small changes in the control and trajectory variables on the terminal constraints. These effects are provided by solving a set of equations which are adjoint to the linear perturbation equations written about a nominal trajectory. A derivation of the properties of the solution of the adjoint equations is given here.

Assume a two-variable system which is described by the non-linear differential equations

$$\dot{x} = f\,(x,\, y,\, t, u(t)) \tag{1}$$

$$\dot{y} = g\,(x,\, y,\, t,\, u(t)) \tag{2}$$

where x and y are the dependent variables, t is the independent variable and u is the control variable. Assume that a solution to these equations is given by the solid line in the figure. One is interested in determining the effect of perturbations δx_i, δy_i and δu on a function Z (x, y) at the terminal time t_f. δx_i and δy_i are known at a particular time t_i and δu is a function of time which is known from t_i to t_f.



The straightforward way to solve this problem is to obtain the solution to Eqs. (1) and (2) with initial conditions

$$x_i = x_{n_i} + \delta x_i$$

$$y_i = y_{n_i} + \delta y_i$$

and a new control variable

$$u = u_n + \delta u$$

where the subscript n denotes the nominal value. The values of x and y at the terminal time are then substituted into Z to determine $\delta Z = Z(x, y) - Z_n(x_n, y_n)$.

If the deviations from the nominal trajectory are small, this process is equivalent to solving the set of linear perturbation equations

$$\delta \dot{x} = \frac{\partial f}{\partial x}\delta x + \frac{\partial f}{\partial y}\delta y + \frac{\partial f}{\partial u}\delta u \qquad (3)$$

$$\delta \dot{y} = \frac{\partial g}{\partial x}\delta x + \frac{\partial g}{\partial y}\delta y + \frac{\partial g}{\partial u}\delta u \qquad (4)$$

Note that if one wanted to change the time t_i at which the perturbations are known, it would be necessary to obtain a new solution to Eqs. (3) and (4) in order to find the terminal perturbations. Thus, it would be very tedious to determine the effects of perturbations at all times t from t_i to t_f. However, by solving the equations which are adjoint to Eqs. (3) and (4), it is possible to obtain the desired information with just one solution of a set of differential equations.

To derive the adjoint equations, begin with the form of the desired answer.

$$\delta Z\Big|_{t_f} = \left[\lambda_x \delta x + \lambda_y \delta y\right]\Big|_t + P(\delta u, t) \qquad t_i \leq t \leq t_f \tag{5}$$

λ_x and λ_y are functions of time which relate perturbations in x and y to the perturbation in Z at the final time. P is an unknown function which represents the influence of δu on $\delta Z|_{t_f}$.

Consider the perturbed trajectory represented by the dotted line in the figure. At time t_1, all the quantities on the right hand side of Eq. (5) will have some value. At t_2, they will have, in general, a slightly different value. However, $\delta Z|_{t_f}$ is always the same for a given perturbed trajectory. Therefore, the quantities on the right hand side of Eq. (5) must change in such a way that $\delta Z|_{t_f}$ remains constant. The time derivative of $\delta Z|_{t_f}$ is zero.

$$\delta \dot{Z}\Big|_{t_f} = 0 = \dot{\lambda}_x \delta x + \lambda_x \delta \dot{x} + \dot{\lambda}_y \delta y + \lambda_y \delta \dot{y} + \dot{P} \tag{6}$$

Substituting Eqs. (3) and (4) into (6) gives

$$0 = \delta x\left(\dot{\lambda}_x + \lambda_x \frac{\partial f}{\partial x} + \lambda_y \frac{\partial g}{\partial x}\right) + \delta y\left(\dot{\lambda}_y + \lambda_x \frac{\partial f}{\partial y} + \lambda_y \frac{\partial g}{\partial y}\right)$$
$$+ \left[\dot{P} + \delta u\left(\lambda_x \frac{\partial f}{\partial u} + \lambda_y \frac{\partial g}{\partial u}\right)\right] \tag{7}$$

λx, λy, and P depend only on the nominal trajectory. One is therefore free to pick any perturbed trajectory which will produce the desired results. In particular, assume that δu is zero and that at some arbitrary time, $\delta y = 0$.

δx is the only remaining perturbation. In order to satisfy Eq. (7), the coefficient of δx must equal zero. Similarly, if δx is assumed to be zero while δy has some value, Eq. (7) is satisfied only if the coefficient of δy is zero. One is therefore led to the differential equations.

$$\dot{\lambda}_x = -\lambda_x \frac{\partial f}{\partial x} - \lambda_y \frac{\partial g}{\partial x} \tag{8}$$

$$\dot{\lambda}_y = -\lambda_x \frac{\partial f}{\partial y} - \lambda_y \frac{\partial g}{\partial y} \tag{9}$$

Eqs. (8) and (9) are adjoint to the linear perturbation equations (Eqs. (3) and (4)) with the forcing function δu set to zero. λx and λy are referred to as adjoint variables. One solution of the adjoint equations provides the effect of perturbations at any time t on $\delta Z|_{t_f}$.

To solve Eqs. (8) and (9), a set of initial conditions is required. These initial conditions are specified at the terminal time, because it is at this point that values are known for λx and λy. Referring to Eq. (5), it is seen that at t_f

$$\lambda_x = \left.\frac{\partial Z}{\partial x}\right|_{t_f} \qquad \lambda_y = \left.\frac{\partial Z}{\partial y}\right|_{t_f}$$

For Z = x,

$$\lambda_x = 1 \qquad \lambda_y = 0$$

Note that the adjoint equations do not depend on Z. Only the initial conditions depend on the form of the terminal constraint.

Now consider a perturbed trajectory for which δu is not zero. The terms in parentheses in Eq. (7) have been shown to be zero. The term in square brackets must also be zero. Therefore

$$\dot{P} = -\left(\lambda_x \frac{\partial f}{\partial u} + \lambda_y \frac{\partial g}{\partial u}\right)\delta u \qquad (10)$$

Integrating Eq. (10) from t_f to t

$$P\Big|_t - P\Big|_{t_f} = -\int_{t_f}^{t} \delta u\left(\lambda_x \frac{\partial f}{\partial u} + \lambda_y \frac{\partial g}{\partial u}\right)dt$$

At the final time t_f, a change in the control, δu, can have no effect on the terminal constraint. P_{t_f} is therefore zero, and

$$P\Big|_t = -\int_{t_f}^{t} \delta u\left(\lambda_x \frac{\partial f}{\partial u} + \lambda_y \frac{\partial g}{\partial u}\right)dt \qquad (11)$$

Thus, to find the influence of changes in the control variable, one evaluates the integral of Eq. (11), where λx and λy are solutions of Eqs. (8) and (9) and the partial derivatives are evaluated along the nominal trajectory.

One is often interested in finding the perturbation in Z at the time that another function S(χ, y) reaches a certain value. S is referred to as the stopping condition. Let T be the unknown time at which the desired value of S is reached. Assuming that T is close to t_f, one may write

$$\delta S\Big|_T = \delta S\Big|_{t_f} + \dot{S}\Big|_{t_f}\delta t \qquad (12)$$

where $\delta t = T - t_f$. $\delta S|_T$ must equal zero if the stopping condition is to be met. Therefore

$$\delta t = -\left.\frac{\delta S}{\dot{S}}\right|_{t_f} \tag{13}$$

Similarly,

$$\delta z|_T = \delta z|_{t_f} + \dot{z}|_{t_f}\,\delta t \tag{14}$$

Substituting Eq. (13) in Eq. (14) gives

$$\delta z|_T = \delta z|_{t_f} - \left.\frac{\dot{z}}{\dot{S}}\right|_{t_f} \delta S|_{t_f} \tag{15}$$

δz and δS at t_f are given by

$$\delta z = \frac{\partial z}{\partial x}\delta x + \frac{\partial z}{\partial y}\delta y \tag{16}$$

$$\delta S = \frac{\partial S}{\partial x}\delta x + \frac{\partial S}{\partial y}\delta y \tag{17}$$

Substituting (16) and (17) into (15) gives

$$\delta z|_T = \left.\left(\frac{\partial z}{\partial x} - \frac{\dot{z}}{\dot{S}}\frac{\partial S}{\partial x}\right)\right|_{t_f} \delta x|_{t_f} + \left.\left(\frac{\partial z}{\partial y} - \frac{\dot{z}}{\dot{S}}\frac{\partial S}{\partial y}\right)\right|_{t_f} \delta y|_{t_f} \tag{18}$$

Referring to Eq. (5) with $t = t_f$, it is seen that λx and λy should be given the following values at t_f.

$$\lambda_x = \left(\frac{\partial Z}{\partial x} - \frac{\dot{Z}}{\dot{S}}\frac{\partial S}{\partial x}\right)\Bigg|_{t=t_f}$$

$$\lambda_y = \left(\frac{\partial Z}{\partial y} - \frac{\dot{Z}}{\dot{S}}\frac{\partial S}{\partial y}\right)\Bigg|_{t=t_f}$$

With these initial conditions, the solution of the adjoint equations will determine the effects of perturbations in x and y on Z at the unknown time when S reaches a desired value.

DERIVATION OF OPTIMIZATION EQUATIONS

Define the following matrices:

$$d\psi = \begin{bmatrix} d\psi_1 \\ d\psi_2 \\ \cdot \\ \cdot \\ d\psi_{jc} \end{bmatrix}$$

where $d\psi_1$ is the desired change in terminal mass at the time the stopping parameter is reached and $d\psi_i$, $i = 2, jc$, are the desired changes in the constraints at that time. jc is the number of terminal constraints, including mass.

$$\Lambda = \begin{bmatrix} \Lambda_{\eta_1} & \Lambda_{\chi_1} \\ \Lambda_{\eta_2} & \Lambda_{\chi_2} \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ \Lambda_{\eta_{jc}} & \Lambda_{\chi_{jc}} \end{bmatrix}$$

Where $\Lambda_{\eta_1} = -\left[\lambda_{v_1}\frac{\partial F}{\partial \eta} + \lambda_{\gamma_1}\frac{\partial G}{\partial \eta} + \lambda_{\psi_1}\frac{\partial H}{\partial \eta}\right]$

$\Lambda_{\chi_1} = -\left[\lambda_{v_1}\frac{\partial F}{\partial \chi} + \lambda_{\gamma_1}\frac{\partial G}{\partial \chi} + \lambda_{\psi_1}\frac{\partial H}{\partial \chi}\right]$

$$\lambda = \begin{bmatrix} \lambda_{v_1} & \lambda_{\gamma_1} & \lambda_{r_1} & \lambda_{\tau_1} & \lambda_{m_1} & \lambda_{\psi_1} & \lambda_{\lambda_1} \\ \lambda_{v_2} & \lambda_{\gamma_2} & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \lambda_{v_{jc}} & \lambda_{\gamma_{jc}} & \lambda_{r_{jc}} & \lambda_{\tau_{jc}} & \lambda_{m_{jc}} & \lambda_{\psi_{jc}} & \lambda_{\lambda_{jc}} \end{bmatrix}$$

where the $\lambda_{(\ )_i}$'s are the solution of the i^{th} set of adjoint equations

$$\delta x = \begin{bmatrix} \delta_V \\ \delta_\gamma \\ \delta_r \\ \delta_{\tau^*} \\ \delta_m \\ \delta_\psi \\ \delta_\lambda \end{bmatrix}$$

where δV = V on present trajectory - V on nominal trajectory at the same time.

δ_{τ}^* is the deviation in inertial longitude

$$W = \begin{bmatrix} 1 & 0 \\ 0 & w \end{bmatrix}$$ where w is a weighting function for the second control variable (χ)

$$Y = \begin{bmatrix} y_1 & & & & 0 \\ & y_2 & & & \\ & & y_3 & & \\ & & & \ddots & \\ 0 & & & & y_m \end{bmatrix}$$ where y_1 is the weighting constant for the first adjustable parameter

$$\delta\alpha = \begin{bmatrix} \delta\eta \\ \delta\chi \end{bmatrix}$$ $\delta\eta$ is the change in the pitch plane component of thrust attitude and $\delta\chi$ is the change in the yaw plane component

$$S = \begin{bmatrix} S_{11} & S_{12} & \cdot & \cdot & \cdot & S_{1_m} \\ S_{21} & \cdot & \cdot & \cdot & \cdot & S_{2_m} \\ S_{31} & \cdot & & & & \\ \cdot & & & & & \\ \cdot & & & & & \\ S_{jc1} & \cdot & \cdot & \cdot & \cdot & S_{jcm} \end{bmatrix}$$ where S_{i_k} is the effect of the k^{th} adjustable parameter on the i^{th} constraint

$$\delta\tau = \begin{bmatrix} \delta\tau_1 \\ \delta\tau_2 \\ \cdot \\ \cdot \\ \delta\tau_m \end{bmatrix}$$ where $\delta\tau_i$ is the change in the i^{th} adjustable parameter

The effects of changes in the trajectory variables, control variables, and adjustable parameters on the terminal quantities is given by

$$d\Psi = \lambda\delta x + \int_{t_f}^{t_i} \Lambda\delta\alpha\, dt + S\delta\tau \qquad (1)$$

One desires to determine the value of $\delta\alpha(t)$ and $\delta\tau$ that will permit (1) to be satisfied for a given $d\Psi$. There is an infinity of solutions to this problem. In order to determine a unique solution, one further requires that the solution to Eq. (1) maximize the following quantity, i.e., make Q, which is a negative number, as large as possible

$$Q = \int_{t_f}^{t_i} \delta\alpha^T W \delta\alpha \, dt - \delta\tau^T Y \delta\tau \tag{2}$$

The superscript T stands for transpose.

The problem, then, is to find $\delta\alpha$ and $\delta\tau$ which maximize Q, while satisfying the constraint on $d\Psi$. Making use of Lagrange multipliers form the quantity

$$V = Q + \mu^T (d\Psi - \lambda \delta x - \int_{t_f}^{t_i} \Lambda \delta\alpha \, dt - S\delta\tau) \tag{3}$$

Combining terms gives

$$V = \int_{t_f}^{t_i} \left[\delta\alpha^T W \delta\alpha - \mu^T \Lambda \delta\alpha\right] dt + \mu^T (d\Psi - \lambda \delta x) - \delta\tau^T Y \delta\tau - \mu^T S \delta\tau \tag{4}$$

Note that V is equal to Q. The $\delta\alpha$ and $\delta\tau$ that maximize V will also maximize Q. δV must equal zero for arbitrary changes in $\delta\alpha$ and $\delta\tau$.

$$\begin{aligned} \delta V = 0 = & \int_{t_f}^{t_i} \left[\delta\alpha^T W \, \delta(\delta\alpha) + \delta(\delta\alpha^T) W \delta\alpha - \mu^T \Lambda \, \delta(\delta\alpha)\right] dt \\ & - \delta\tau^T Y \, \delta(\delta\tau) - \delta(\delta\tau)^T Y \delta\tau - \mu^T S \, \delta(\delta\tau) \end{aligned} \tag{5}$$

$\delta(\delta\alpha^T) W \delta\alpha$ is a one-by-one matrix and is therefore equal to its transpose $\delta\alpha^T W^T \delta(\delta\tau)$. Also, $W = W^T$. Therefore,

$$\delta V = 0 = \int_{t_f}^{t_i} \left[2\delta\alpha^T W - \mu^T \Lambda\right] \delta(\delta\alpha) \, dt - (2\delta\tau^T Y + \mu^T S) \, \delta(\delta\tau) \tag{6}$$

The coefficients of $\delta(\delta\alpha)$ and $\delta(\delta\tau)$ must equal zero if δV is to be zero for changes in $\delta\alpha$ and $\delta\tau$. Therefore,

$$\delta\alpha^T = 0.5 \, \mu^T \Lambda W^{-1} \quad \text{or} \quad \delta\alpha = 0.5 \, W^{-1} \Lambda^T \mu \tag{7}$$

and

$$\delta\tau^T = -0.5 \, \mu^T S Y^{-1} \quad \text{or} \quad \delta\tau = -0.5 \, Y^{-1} S^T \mu \tag{8}$$

To solve for μ, substitute Eqs. (7) and (8) into (1).

$$d\Psi = \lambda\,\delta x + \frac{1}{2}\left[\int_{t_f}^{t_i} \Lambda W^{-1} \Lambda^T dt\right]\mu - \frac{1}{2} S Y^{-1} S^T \mu \tag{9}$$

and

$$\mu = 2\left[\int_{t_f}^{t_i} \Lambda W^{-1} \Lambda^T dt - S Y^{-1} S^T\right]^{-1} \left[d\Psi - \lambda\,\delta x\right] \tag{10}$$

Substitute back into Eqs. (7) and (8) to give

$$\delta\alpha = W^{-1}\Lambda^T\left[\int_{t_f}^{t_i} \Lambda W^{-1} \Lambda^T dt - S Y^{-1} S^T\right]^{-1} (d\Psi - \lambda\,\delta x) \tag{11}$$

$$\delta\tau = -Y^{-1} S^T\left[\int_{t_f}^{t_i} \Lambda W^{-1} \Lambda^T dt - S Y^{-1} S^T\right]^{-1} (d\Psi - \lambda\,\delta x) \tag{12}$$

At this point it is useful to indicate the difference between the PRESTO optimization procedure and that of previous programs.[1,2] In Refs. 1 and 2, the matrix in the brackets in Eqs. (11) and (12) is inverted only once at the initial time t_i. The inverted matrix is then multiplied by the vector $d\Psi$ (δx is zero, assuming initial conditions to be fixed) to form a vector of constants. This constant vector is then multiplied by $W^{-1}\Lambda^T$, which is a function of time to determine the entire $\delta\alpha$ time history for the trajectory. $\delta\tau$ is evaluated in the same manner with $Y^{-1}S^T$ used instead of $W^{-1}\Lambda^T$. Thus, the changes in the control and the adjustable parameters are fixed at the beginning of the forward trajectory. In control system terminology, this is an open-loop system.

In the PRESTO program, every point is treated as the initial point. The time t_i becomes the running variable t and the bracketed matrix is inverted

---

[1] Kelley, H. J., "Gradient Theory of Optimal Flight Paths," ARS Journal 30, 947-953 (1960).

[2] Bryson, A.E., and Denham, W. F., "A Steepest-Ascent Method for Solving Optimum Programming Problems," J. Applied Mechanics 29, 247-257 (1962).

at every point of integration while running a backward trajectory. During forward trajectories, the change in the control to be used for the remainder of the trajectory is recomputed at each point, taking into account the deviation from the nominal trajectory at that point. The program operates in a closed-loop fashion because it continuously checks how it is doing in its attempt to satisfy terminal conditions.

The advantage of this closed-loop approach is that larger deviations from the nominal trajectory can be tolerated while still meeting terminal conditions. It is, therefore, possible to move more rapidly from the initial nominal trajectory to the optimum trajectory.

It is not possible to use the closed-loop mode of operation for the entire trajectory. Near the end of the trajectory the control variables become too sensitive to small perturbations from the nominal. It is, therefore, necessary to switch back to open-loop operation at some point on the trajectory.

Selection of Initial Improvement in Terminal Mass

In general, one does not know how far from the optimum a given nominal trajectory will be. It is, therefore, difficult to guess how much mass improvement to ask for. On the other hand, one can predict reasonable values for the expected changes in control variables and adjustable parameters. The procedure to be used, then, is to guess changes in the controls and let the computer determine the corresponding change in terminal mass. The required calculations are given here.

Define dP^2 as

$$dP^2 = \int_{t_f}^{t_i} \delta\alpha^T W \delta\alpha \, dt - \delta\tau^T Y \delta\tau \qquad (13)$$

dP^2 is a measure of the amount of control change. To obtain an estimate of dP^2, average values of $\delta\eta$ and $\delta\chi$ are selected. These are squared and multiplied by the expected burn time to get an approximate value for the integral. Reasonable changes in the adjustable parameters are also selected and their squares, modified by the weighting functions, are added to the integral terms to determine dP^2.

Substitute Eqs. (11) and (12) with $\delta X = 0$, into Eq. (13) to obtain dP^2 in terms of the changes in the terminal constraints.

$$dP^2 = d\Psi^T \left[\int_{t_f}^{t_i} \Lambda W^{-1} \Lambda^T dt - S Y^{-1} S^T \right]^{-1} d\Psi \qquad (14)$$

Let the inverted matrix be denoted as A with components Aij. Split the d vector into two parts, δm_d and $d\phi$.

$$d\Psi = \begin{bmatrix} \delta m_d \\ d\Phi \end{bmatrix}$$

where δm_d is the initial change in terminal mass and

$$d\phi = \begin{bmatrix} d\Psi_2 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ d\Psi_{jc} \end{bmatrix}$$

The object now is to solve Eq. (14) for δm_d in terms of dP^2 and $d\phi$. Rewrite Eq. (14) as

$$dP^2 = \begin{bmatrix} \delta m_d & d\phi \end{bmatrix} \begin{bmatrix} A_{11} & A_{12} & \cdot & \cdot & \cdot & \cdot & \cdot \\ A_{21} & A_{22} & A_{23} & & & & \\ \cdot & A_{32} & \cdot & & & & \\ & & & & & & \end{bmatrix} \begin{bmatrix} \delta m_d \\ d\Phi \end{bmatrix}$$

Let the minor of A_{11} be designated as

$$M = \begin{bmatrix} A_{22} & A_{23} & \cdot & \cdot & \cdot \\ A_{32} & \cdot & \cdot & \cdot & \cdot \\ \cdot & & & & \\ \cdot & & & & \end{bmatrix}$$

Then $dP^2 = A_{11} \; \delta m_d^{\;2} + 2N^T d\phi \, \delta m_d + d\phi^T M d\phi$ (15)

where

$$N = \begin{bmatrix} A_{21} \\ A_{31} \\ \cdot \\ \cdot \end{bmatrix}$$

Solving Eq. (15) for δm_d gives

$$\delta m_d = \frac{-N^T \, d\phi}{A_{11}} + \sqrt{\frac{dP^2 - d\phi^T \left(M - \frac{NN^T}{A_{11}}\right) d\phi}{A_{11}}} \tag{16}$$

PROGRAMMING OF THE OPTIMIZATION EQUATIONS

The programming of Eqs. (11) and (12) of the previous section will be described here. The quantities that are to be stored differ depending on whether the computation is in the closed-loop or open-loop mode. To indicate the difference, Eqs. (11) and (12) are rewritten in the following manner.

Closed Loop

$$\delta\alpha = B\, d\Psi - D\, dX \tag{1}$$

$$\delta\tau = C\, d\Psi - E\, dX \tag{2}$$

where

$$B = W^{-1}\Lambda^T A$$

$$A = \left[\int_{t_f}^{t_i} \Lambda W^{-1}\Lambda^T dt - S\, Y^{-1} S^T\right]^{-1}$$

$$D = B\,\lambda$$

$$C = -Y^{-1} S^T A$$

$$E = C\,\lambda$$

Open Loop

$$\delta\alpha = W^{-1}\Lambda^T K \tag{3}$$

$$\delta T = -Y^{-1} S^T K \tag{4}$$

where $K = A\left[d\Psi - \lambda\, dX\right]$

Note that K is evaluated only once, at the point that the computation switches from closed to open loop operation.

Terminology

A forward run goes forward in time. With the exception of the first forward run, all forward runs are either guidance or optimizations runs. On a forward guidance run, one attempts only to meet terminal conditions. On a forward optimization run, one attempts to obtain mass improvement along with meeting terminal conditions. The first forward run (the initial trajectory) uses a control program that is read in.

A backward run goes backward in time. All backward runs are either guidance or optimization runs depending on whether the following forward run is to be a guidance or optimization run. The equations for the quantities which are to be stored on the backward run are different for guidance and optimization.

A count is made of the number of integration steps. KPOINL is the number of the integration step at which the computation switches from the closed-loop to the open-loop mode of operation.

The number of constraints, including mass, is given by JC. IC is 1 for optimization runs and 2 for guidance runs.

Computations to be made on Backward Runs

$$\left.\begin{aligned} \Lambda_{\eta_j} &= -\left[\lambda_{V_j}\frac{\partial F}{\partial \eta} + \lambda_{\gamma_j}\frac{\partial G}{\partial \eta} + \lambda_{\psi_j}\frac{\partial H}{\partial \eta}\right] && (5) \\ \Lambda_{\chi_j} &= -\left[\lambda_{V_j}\frac{\partial F}{\partial \chi} + \lambda_{\gamma_j}\frac{\partial G}{\partial \chi} + \lambda_{\psi_j}\frac{\partial H}{\partial \chi}\right] && (6) \end{aligned}\right\} j = 1, JC$$

The Λ's are stored from the final point to KPOINL. They are placed in the DD storage area. The indexes ML1 and ML2 are the row locations in storage for the two columns of the Λ matrix being stored at the current integration step. ML1 and ML2 start at 303 and 304, respectively, and are reduced by two at each integration step until KPOINL is reached. Then, ML1 and ML2 are set equal to 1 and 2. During the closed-loop portion of the trajectory, the current Λ matrix is stored in the first two rows of DD.

$$I_{ij} = \int_{t_f}^{t} \left[\Lambda_{\eta_i}\Lambda_{\eta_j} + \frac{\Lambda_{\chi_i}\Lambda_{\chi_j}}{w}\right] dt \qquad (7)$$

for all i, j from 1 to jc for $i \leq j$

$$J_{ij} = \sum \frac{S_{i_K} S_{j_K}}{Y_K} \qquad i, j = 1, \ldots JC \quad \text{for } i \leq j \qquad (8)$$

J_{ij} is computed only for the values of k corresponding to adjustable parameters that are being optimized. The following table relates the value of k to a particular adjustable parameter.

k	Adjustable Parameter
1	Length of coast after second burn in stage 4
2	Length of second burn in stage 4
3	Length of coast after first burn in stage 4
4	Length of first burn in stage 4
5	Length of coast after stage 3
6	Length of coast after stage 2
7	Length of coast after stage 1
8	Time of day of launch
9	Initial azimuth angle
10	Initial flight path angle

S_{ik} is computed as follows:

for $1 \leq k \leq 8$

$$S_{ik} = \lambda P$$

where

$$P = \begin{bmatrix} \dot{V} \\ \dot{\gamma} \\ \dot{r} \\ \dot{T} + \omega \\ \dot{m} \\ \dot{\psi} \\ \dot{\lambda} \end{bmatrix}$$

P is evaluated at the end of the coast or burn that is being optimized.

for k = 9 and 10

$$S_{i9} = \begin{bmatrix} \lambda_{\psi_1} \\ \lambda_{\psi_2} \\ \vdots \\ \lambda_{\psi_{JC}} \end{bmatrix} \qquad S_{i10} = \begin{bmatrix} \lambda_{\gamma_1} \\ \lambda_{\gamma_2} \\ \vdots \\ \lambda_{\gamma_{JC}} \end{bmatrix}$$

I_{ij} and J_{ij} are added term by term. The matrix $(I_{ij} + J_{ij})$ is then inverted. On backward guidance runs, the first row and column of $(I_{ij} + J_{ij})$ are not included in the matrix to be inverted. On backward optimization runs, the entire matrix is inverted.

Let the inverted matrix be represented by A. Store A when KPOINC = KPOINL. It is known as A92 at that point.

$$B_{1i} = \sum_{j = IC}^{JC} \Lambda_{\eta j} A_{ij} \qquad i = IC, \ldots\ldots\ldots\ldots JC \qquad (9)$$

$$B_{2i} = \sum_{j = IC}^{JC} \Lambda_{xj} A_{ij} \qquad i = IC, \ldots\ldots\ldots\ldots JC \qquad (10)$$

$$B_{12}^{*} = \sum_{j = 2}^{JC} B_{1j}\, d\Psi_j \qquad (11)$$

$$B_{22}^{*} = \sum_{j = 2}^{JC} B_{2j}\, d\Psi_j \qquad (12)$$

Store B_{11} and B_{21} in the first column of B matrix and B_{12}^{*} and B_{22}^{*} in the second column. Two rows of the B matrix are stored at each integration point. The indexes MB1 and MB2 give the row location of the matrix being stored at the current time. The B's are stored only during the closed-loop part of the trajectory.

$$C_{ki} = - \sum_{j = IC}^{JC} \frac{S_{jk} A_{ij}}{y_k} \qquad i = IC, \ldots\ldots\ldots JC \qquad (13)$$

The C's are computed and stored, at one time point only, after each adjustable parameter, or group of parameters, is introduced. The number of rows of the C matrix that are stored at any one time depends on the number of parameters remaining to be adjusted from that time forward to the end of the trajectory. The indexes MC1 and MC2 represent the first and last row of the matrix being stored at the current point.

$$D = B\lambda \tag{14}$$

D is a 2 x 7 matrix. It is stored in the DD storage area. The indexes MB1 and MB2 give the row location of the D matrix being stored at the current time. D is stored only during the closed-loop part of the trajectory.

$$E = C\lambda \tag{15}$$

E has the same number of rows as does C. It is computed and stored at the same time as C and uses the same indexes, MC1 and MC2.

<u>Computation to be made on all forward runs after the first</u>

For KPOINC $<$ KPOINL (closed-loop)

$$\delta\eta = B_{11} d\Psi_1 + B_{12}^* - \sum_{\ell=1}^{7} D_{1\ell}\,\delta x_\ell \tag{16}$$

$$\delta\chi = B_{21} d\Psi_1 + B_{22}^* - \sum_{\ell=1}^{7} D_{2\ell}\,\delta x_\ell \tag{17}$$

$$\delta\tau_K = \sum_{j=IC}^{C} C_{kj} d\Psi_j - \sum_{\ell=1}^{7} E_{k\ell}\,\delta x_\ell \tag{18}$$

For KPOINC = KPOINL

Compute $K = A\left[d\dot{\Psi} - \lambda\,\delta x\right]$

where A and λ have been stored at this point on the last backward run.

For KPOINC $\geq$ KPOINL

$$\delta\dot{\eta} = \sum_{j=IC}^{JC} K_j \Lambda_{\eta j} \tag{19}$$

$$\delta\chi = \frac{1}{w} \sum_{j=IC}^{JC} K_j \Lambda_{\chi j} \tag{20}$$

$$\delta\tau_K = \frac{1}{\bar{y}_k} \sum_{j=IC}^{JC} K_j S_{jk} \tag{21}$$

Logic for Determining Sequence of Runs

The first run is forward with the trajectory variables for the following backward guidance run stored. The second run is a backward guidance run. The third run is a forward guidance run. After each forward run, the quantities $d\psi_i$ ($i = 2, \ldots, JC$) are computed. A check is then made of the magnitude (absolute value) of $d\psi_i$ compared to permitted values of the deviations, $d\psi_{pi}$, which will be read in. If

$$|d\psi_i| < d\psi_{pi} \quad \text{for all } i \;, \quad i = 2, \ldots, JC$$

then a backward optimization run is made. If $|d\psi_i| > d\psi_{pi}$ for any i , then a backward guidance run is made. A forward guidance run follows in an attempt to satisfy the check on terminal constraints. If satisfied, a backward optimization run is made. If not, another set of backward and forward guidance runs is made. This process continues until the constraint check is satisfied, after which a backward optimization run is made.

At the end of the backward optimization run, δm_d is computed. If the square root is real and δm_d is positive, a forward optimization run is made. If the square root is complex or if δm_d is negative, an input value of δm_d is used. A forward optimization run is then made. At the end of the forward optimization run, the following checks are made:

(1) is $\delta m|_{tf}$ positive, where $\delta m|_{tf}$ = m at end of run - m at end of last forward run.

If the answer to this is yes,

(2) is $|d\psi_i| < d\psi_{pi}$ for all i

If the answer to this is yes, a backward optimization run is made followed by another forward optimization run.

If the answer to either question is no, then δm_d is cut in half and another forward optimization run is made. This continues until the answers to questions (1) and (2) are both yes.

Eventually, the improvement asked for in mass, δm_d, becomes less than ϵ. At this point a backward guidance run is made, followed by a final forward guidance run.

The logic is illustrated in the following diagram.

Discontinuity in Adjoint Variables Across Closed-Form Coast Periods

The perturbations in trajectory variables are discontinuous across a closed-form coast. Consequently, a discontinuity must be introduced into the adjoint variables so that

$$d\Psi = \lambda^1 \delta x^1 = \lambda^2 \delta x^2 \qquad (22)$$

where the superscripts 1 and 2 represent the beginning and end of the coast, respectively.

The perturbations at the end of the coast are expressed in terms of the initial perturbations through the equation

$$\delta x^2 = \frac{\delta x_2}{\delta x_1} \delta x^1 \qquad (23)$$

Substitute Eq. (23) into (22) to give

$$\lambda^1 \delta x^1 = \lambda^2 \frac{\delta x_2}{\delta x_1} \delta x^1 \qquad (24)$$

The coefficients of δx^1 on both sides of Eq. (24) must be equal. Therefore

$$\lambda^1 = \lambda^2 \frac{\partial x_2}{\partial x_1} \qquad (25)$$

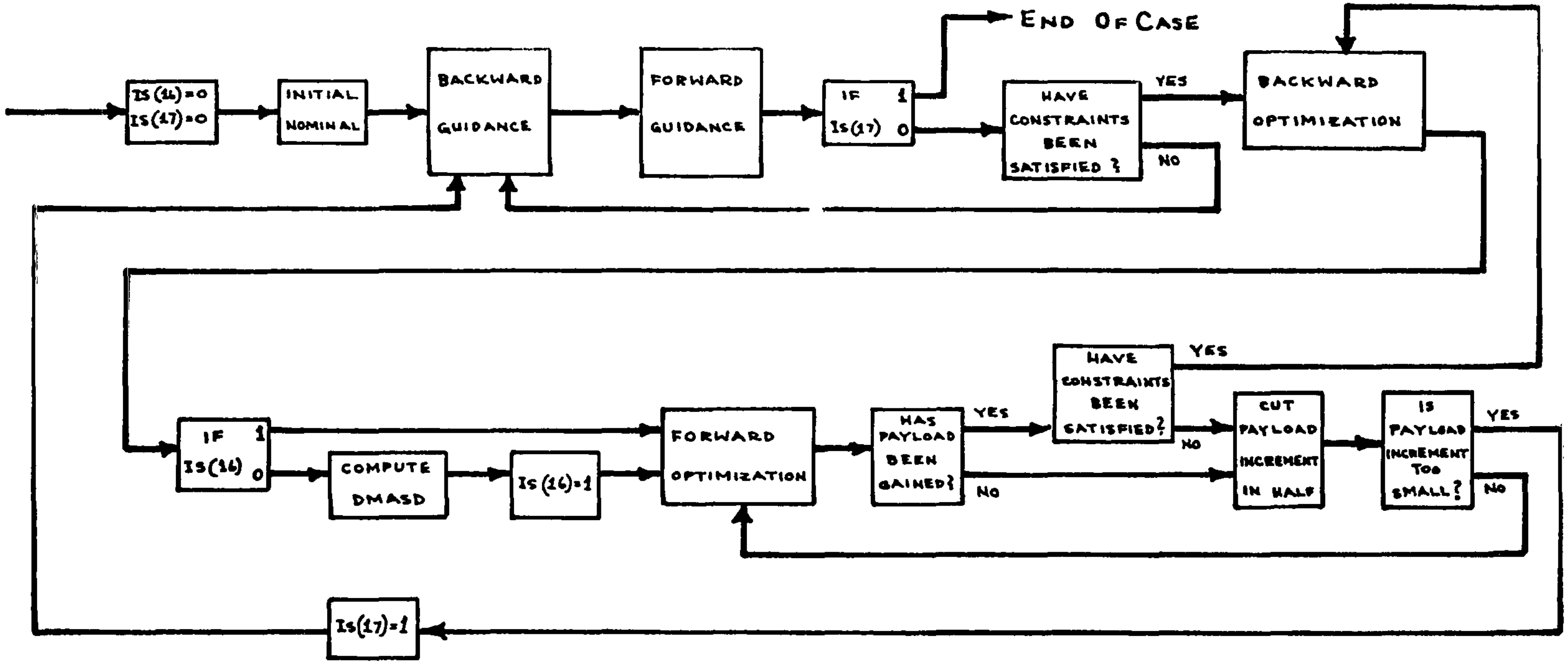
The matrix of partial derivatives is

$$\frac{\partial x_2}{\partial x_1} = \begin{bmatrix} \frac{\partial V_2}{\partial V_1} & \frac{\partial V_2}{\partial \gamma_1} & \frac{\partial V_2}{\partial r_1} & 0 & 0 & \frac{\partial V_2}{\partial \Psi_1} & \frac{\partial V_2}{\partial \lambda_1} \\ \frac{\partial \gamma_2}{\partial V_1} & \frac{\partial \gamma_2}{\partial \gamma_1} & \frac{\partial \gamma_2}{\partial r_1} & 0 & 0 & \frac{\partial \gamma_2}{\partial \Psi_1} & \frac{\partial \gamma_2}{\partial \lambda_1} \\ \frac{\partial r_2}{\partial V_1} & \frac{\partial r_2}{\partial \gamma_1} & \frac{\partial r_2}{\partial r_1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & \frac{\partial \tau_2}{\partial \Psi_1} & \frac{\partial \tau_2}{\partial \lambda_1} \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ \frac{\partial \Psi_2}{\partial V_1} & \frac{\partial \Psi_2}{\partial \gamma_1} & \frac{\partial \Psi_2}{\partial r_1} & 0 & 0 & \frac{\partial \Psi_2}{\partial \Psi_1} & \frac{\partial \Psi_2}{\partial \lambda_1} \\ \frac{\partial \lambda_2}{\partial V_1} & \frac{\partial \lambda_2}{\partial \gamma_1} & \frac{\partial \lambda_2}{\partial r_1} & 0 & 0 & \frac{\partial \lambda_2}{\partial \Psi_1} & \frac{\partial \lambda_2}{\partial \lambda_1} \end{bmatrix}$$

These partial derivatives are evaluated as indicated in the next section.

Derivatives Across Closed-Form Coast

For the solution of adjoint equations across a coast trajectory, an evaluation is required of partial derivatives of the state variables at the end of coast with respect to state variables at the beginning of coast and with respect to the (time) duration of coast. For the coast stages being computed using closed-form equations, these partials are evaluated numerically by using the coast equations. That is, a nominal calculation of the coast provides a set of state variables at beginning and end of the coast. Then, by entering the coast equations again with an incremented value of initial velocity (e.g.) a new (perturbed) set of state variables at the end of coast is obtained. The partials of the state variables with respect to initial velocity are then formed by a simple ratio of differences obtained in the two calculations. This procedure is used for $\frac{\partial(V_2, r_2, \gamma_2, \lambda_2, \tau_2, \Psi_2)}{\partial(V_1, r_1, \gamma_1, \lambda_1, \tau_1, \Psi_1)}$. It includes transformations from rotating to inertial at beginning of coast and from inertial to rotating at the end.



FLOW DIAGRAM ILLUSTRATING SEQUENCE OF RUNS

The partials of state variables at end of coast with respect to the coast duration are formed in the same way as in the integrated coast stage. That is, time derivatives of the trajectory variables at end of coast, with thrust = 0., are obtained from the DEQ subroutine. These are multiplied by the local coast-angle rate-of-change to obtain the rate-of-change of the state variables with respect to coast angle.

Initial Conditions for Integration of the Adjoint Equations

Referring to page 8-7, it is seen that the initial conditions on the adjoint variables are specified at the final time and are given by

$$\lambda_x = \left(\frac{\partial z}{\partial x} - \frac{\dot{z}\,\partial S}{\dot{S}\,\partial x} \right) \Bigg|_{t = t_f}$$

where x is one of the trajectory variables

Z is the terminal constraint

and S is the stopping parameter

In PRESTO the first set of adjoint equations is always used for the mass constraint and the remaining sets for the more general constraints. Thus, for the first set of equations, Z = m, and the initial conditions are given by

$$\lambda V_1 = -\left(\frac{\dot{m}}{\dot{S}}\right)\left(\frac{\partial S}{\partial V}\right) \qquad \lambda m_1 = 1$$

$$\lambda_{\gamma_1} = -\left(\frac{\dot{m}}{\dot{S}}\right)\left(\frac{\partial S}{\partial \gamma}\right) \qquad \lambda_{\Psi_1} = -\frac{\dot{m}}{\dot{S}}\frac{\partial S}{\partial \Psi}$$

$$\lambda_{r_1} = -\frac{\dot{m}}{\dot{S}}\frac{\partial S}{\partial r} \qquad \lambda_{\lambda_1} = -\frac{\dot{m}}{\dot{S}}\frac{\partial S}{\partial \lambda}$$

The initial conditions for the other sets of initial conditions are

$$\lambda_{V_i} = \frac{\partial Z_i}{\partial V} - \frac{\dot{Z}_i}{\dot{S}} \frac{\partial S}{\partial V} \qquad \lambda_{m_i} = 0$$

$$\lambda_{\gamma_i} = \frac{\partial Z_i}{\partial \gamma} - \frac{\dot{Z}_i}{\dot{S}} \frac{\partial S}{\partial \gamma} \qquad \lambda_{\psi_i} = \frac{\partial Z_i}{\partial \psi} - \frac{\dot{Z}_i}{\dot{S}} \frac{\partial S}{\partial \psi}$$

$$\lambda_{r_i} = \frac{\partial Z_i}{\partial r} - \frac{\dot{Z}_i}{\dot{S}} \frac{\partial S}{\partial r} \qquad \lambda_{\lambda_i} = \frac{\partial Z_i}{\partial \lambda} - \frac{\dot{Z}_i}{\dot{S}} \frac{\partial S}{\partial \lambda}$$

In the coding of the ICS subroutine, a great many partial derivatives and total derivatives are formed. The following nomenclature conventions are used.

For partial derivatives the first letter is P, the middle is the dependent variable, and the last letter is the independent variable. Thus PVHVI = $\partial V_H / \partial V_I$.

For total derivatives the first letter is D and the rest is the same. Thus, DVHV = dV_H/dV.

SUBROUTINES

PRESTO ATMOSPHERE SUBROUTINE
(1959 ARDC MODEL)

Required: Density, (ρ) Pressure, (p) Speed of Sound (a)

Symbols

- h — geometric altitude in feet ($h = r - R_e$)
- H* — geopotential altitude in geopotential feet'
- ρ — density in slugs
- p — pressure in lb./ft.^{2}
- a — speed of sound
- T_M — Molecular-scale temperature in °R @ altitude H*
- L_M — gradient of T in terms of H*; i.e., $\frac{\partial T_M}{\partial H^*}$ in °R/ft.'

Subscripts

- o • denotes property at sea level $h = H^* = 0$
- b • denotes property at base of particular layer

Constants

$g_o = 32.154856$ ft/sec^2 (acceleration of gravity measured at sea level)

$R^* = 1715.4827$ ft^2/°Rsec^2, gas constant for air

$R_e = 20{,}902{,}900$ ft., radius of earth

Equations

$$H^* = \frac{R_e \cdot h}{R_e + h}$$

TYPE I $(L_M)_b = 0$

$$T_M = (T_M)_b$$

$$\rho = \rho_b \exp\left[\frac{-g_o(H^* - H^*_b)}{R^*(T_M)_b}\right]$$

$$p = \rho R^*(T_M)_b$$

$$a = \sqrt{1.4R^*(T_M)_b}$$

TYPE II $(L_M)_b \neq 0$

$$T_M = (T_M)_b + (L_M)_b(H^* - H_b^*)$$

$$\rho = \rho_b \left[1 + \frac{(L_M)_b (H^* - H^*_b)}{(T_M)_b}\right]^{-\left(1 + \frac{g_0}{R^*(L_M)_b}\right)}$$

$$p = \rho R^* T_M$$

$$a = \sqrt{1.4 R^* T_M}$$

TABLE OF CONSTANTS AT BASE ALTITUDES

H^*_b	$(T_M)_b$	$(L_M)_b$	ρ_b
0	518.69	-3.56616×10^{-3}	2.3769×10^{-3}
36,089.239	389.988	0	7.0547×10^{-4}
82,020.997	389.988	1.646592×10^{-3}	7.7615×10^{-5}
154,199.475	508.788	0	2.8829×10^{-6}
173,884.514	508.788	-2.46888×10^{-3}	1.3964×10^{-6}
259,186.352	298.188	0	4.1123×10^{-8}
295,275.591	298.188	2.19456×10^{-3}	4.2560×10^{-9}
344,488.189	406.188	1.09728×10^{-2}	2.2243×10^{-10}

DERIVATIVES OF (ρ, p, a) WITH RESPECT TO ALTITUDE

1. Assume $\frac{\partial(\ )}{\partial h} = \frac{\partial(\ )}{\partial H^*}$

2. For TYPE I $(L_M)_b = 0$

$$\frac{\partial \rho}{\partial h} = -\frac{\rho\, g_0}{R^*(T_M)_b}$$

$$\frac{\partial p}{\partial h} = R^*(T_M)_b \frac{\partial \rho}{\partial h}$$

$$\frac{\partial a}{\partial h} = 0$$

3. For TYPE II $(L_M)_b \neq 0$

$$\frac{\partial \rho}{\partial h} = \frac{-\rho \left[1 + \frac{g_o}{R^*(L_M)_b}\right] \left[\frac{(L_M)_b}{(T_M)_b}\right]}{\left[1 + \frac{(L_M)_b (H^* - H^*_b)}{(T_M)_b}\right]}$$

$$\frac{\partial p}{\partial h} = R^* \left(T_M \frac{\partial \rho}{\partial h} + \rho (L_M)_b\right)$$

$$\frac{\partial a}{\partial h} = \frac{0.7\ R^*(L_M)_b}{a}$$

SUBROUTINE LUNEPH

Purpose

The subroutine calculates the position of the moon at a given date.

Input: T = date in "space age date" (number of days since January 0.0, 1960) at which the position of the moon is calculated.

Output: R = position vector (distance in feet, right ascension and declination in radians) of the moon relative to the earth in the inertial coordinate system defined by the Ecliptic 1950.0 and the equator of date.

Calling: CALL LUNEPH (T, R)

Method of Solution: The program utilizes the equations derived in "Development of a Computer Subroutine for Planetary and Lunar Positions," by H. F. Michielsen and M. A. Krop (WADD Technical Report 60-118). The ephemeris is analytic in form, where the lunar distance and two position angles are computed from summations of time-periodic terms.

Accuracy: Maximum errors in computed lunar radial distance and angular position are approximately 75 miles and .05 degrees, respectively.

SUBROUTINE PLANEP

Purpose

The subroutine calculates the hyperbolic excess velocity vector for leaving earth and going to Venus or Mars.

Inputs: T1 = date leaving earth, expressed in "space age date" (number of days since 1960.0).

DELT = trip time from earth to arrival planet in days.

IB = arrival planet number Mars = 2

Venus = 3

Outputs: VH = hyperbolic excess speed vector (magnitude, right ascension, declination) in the inertial coordinate system defined by ecliptic and mean equinox 1960.0.

Calling: CALL PLANEP (TI, DELT, IB, VH)

Method of Solution: The subroutine is an extract from the Lockheed "Medium Accuracy Interplanetary Transfer Program" (S. Ross). This program assumes the planets to move in fixed heliocentric ellipses mutually inclined. Using a modified form of Lamberts Theorem the routine solves for the heliocentric transfer arc connecting the two planets at the specified dates. It then computes the geocentric-referenced magnitude and direction of the heliocentric ellipse velocity at Earth center. This geocentric velocity represents the required magnitude and direction of the departure hyperbolic excess velocity vector. That is, the asymptote of the departure hyperbola must be directed along this vector. An approximation is allowed, however, in that any hyperbola (of proper energy) whose asymptote lies parallel to the required vector is acceptable. Thus, a "conic overlap" construction is employed instead of a matching at a particular point in space. This construction materially simplifies the calculation procedure and comparisons have shown that the accuracy obtained thereby lies well within design tolerances.

THE RKAD INTEGRATION SUBROUTINE

The integration subroutine in PRESTO makes use of both the Runge-Kutta and the Adams methods of integration. The Adams method is faster than the Runge-Kutta but requires knowledge of the variable at a number of points prior to the current point. It, therefore, cannot be used to start the integration. The Runge-Kutta method does not require past information and can be used to start the integration.

The approach used in the program, then, is to start the integration with the Runge-Kutta method and to switch to the Adams method after four steps of integration. This procedure is repeated at the beginning of each stage because the Adams method cannot tolerate the thrust discontinuities associated with staging.

If there are any discontinuities in thrust or rate-of-change of thrust within a stage, it is necessary to use the Runge-Kutta method until the discontinuity has been passed. Provision has been made in the program to keep the integration package in the Runge-Kutta mode until a velocity, specified in the input, is reached. Also, if the change in angle-of-attack exceeds 15 degrees from point to point, the integration will switch to the Runge-Kutta method.

The Runge-Kutta method used is standard. The Adams method computes the increment in any variable in terms of the current derivative and the derivatives at the three previous points. The increment δy is given by

$$\delta y = \left(55\dot{y}_1 - 59\dot{y}_2 + 37\dot{y}_3 - 9\dot{y}_4\right)\frac{\delta t}{24}$$

where δt is the size of the integration step, $\dot{y}_1$ is the current derivative, and $\dot{y}_2$ is the derivative one point back, etc.

** THIS SUBROUTINE HAS BEEN REPLACED BY SUBROUTINE SYMVRT, PAGES 9-9 and 9-10

SUBROUTINE INVERT

July 24, 1964

Purpose

This subroutine calculates the inverse of a matrix.

Method

The method of Crout is used. The matrix A is decomposed into an upper triangular matrix U and a lower triangular matrix L such that $A = L U$. Then L^{-1} and U^{-1} are computed by a closed set of operations. $A^{-1} = U^{-1} L^{-1}$ is then computed. During the decomposition a pivot search is employed following Wilkinson's precriptions for retaining accuracy. For a complete description of the Crout method see F. E. Hildebrand, "Introduction to Numerical Analysis," McGraw-Hill, 1956, pp 429-439.

Usage

Entrance is made via the FORTRAN statement in the calling program,

CALL INVERT (A, IMAX, ISING)

where 1) A is the tag of the matrix to be inverted. After the inversion is complete, the inverse is stored in this tag. Therefore, the original matrix is destroyed.

2) IMAX is the number of rows in the matrix.

3) ISING will be set equal to zero if the inversion was successful. ISING will equal one if the matrix in A was singular.

The above dummy variables may be replaced by other suitable names. For compatibility purposes with this subroutine which was compiled with dimension entries of 6x6, dimension entries for the 2 dimensional arrays in the calling program must have a row dimension equal to 6. All the entries of 6 in the subroutine can be changed to a more suitable value.

Space Required

807_{10} $(1447)_8$ locations

Timing

Running time (T in seconds) is approximately $T = .001n^3$, where n is the number of rows in the matrix.

Restrictions

There are no inherent limits of the size of the matrix to be inverted. Accuracy of the inverse depends upon the particular case. It is recommended that the multiplication $A A^{-1} = I$ to be performed and the identity checked. For a large group of 70 x 70 production cases this check has been made and the identity was observed to be accurate to no less than six decimal places in all cases. A badly conditioned matrix like the Hilbert matrix will fail miserably at 7 x 7.

Author

Ira Hanson, Lockheed Missiles & Space Company
July, 1961

IDENTIFICATION

F1*ML F HSIV SYMMETRIC MATRIX INVERSION
Ira C. Hanson
November 1962

PURPOSE

This subroutine calculates the inverse of a symmetric matrix.

METHOD

The algorithm of Cholesky is used to decompose the symmetric matrix A into a triangular matrix B such that $A = BB^*$. The asterisk denotes transpose. If the matrix A is not positive-definite the matrix B will contain some imaginary elements. The triangular matrix B is then inverted by direct elimination. Since $(B^{-1})^* = (B^*)^{-1}$ it is unnecessary to compute $(B^*)^{-1}$. The final inverse is then computed as follows. $A^{-1} = (B^*)^{-1} B^{-1}$. All imaginary elements drop out at this point. Only the upper triangular part of A is used in the computation.

For a complete description of the algorithm of Cholesky see E. Rodewig, "Matrix Calculus," North Holland Publishing Company-Amsterdam, 1956, pp. 110-114.

USAGE

Entrance to the subroutine is made via the FORTRAN statement in the calling program.

```
CALL SYMVRT (A, N, ISING)
```

where 1) A is the label of the matrix to be inverted. Only the upper triangular part of A is required. After the inversion is complete, the inverse is stored in the lower triangular part of A. The original matrix is destroyed.

2) N is the number of rows in the matrix.

3) ISING will be set to zero if the inversion was successful. ISING will equal one if the matrix in A is singular.

The subroutine uses three temporary single subscripted arrays. These arrays must be dimensioned at least as large as the row entry of the A array. These arrays may be placed in COMMON to conserve storage if desired.

RESTRICTIONS

The Cholesky decomposition will fail if a zero appears during the computation of the diagonal elements and also if A(1,1)=0. This does not necessarily mean the matrix is singular, but it does mean the calculation of the inverse has failed. There is no practical fool-proof method of apriorily interchanging rows and columns to avoid this trouble. Interchanging after a zero is detected is not a solution because the remaining elements may also be zero and the matrix still not be singular. Therefore no pivot search is attempted in this subroutine and it should only be used in a physical application where it is known that this restriction is not prohibitive. Otherwise the Crout method subroutine F1 * ML F HINT should be used which has a complete pivot search.

SPACE REQUIRED

620 cells are required.

TIMING

Running time (T in seconds) is approximately $T = .00005\ n^3$, where n is the number of rows in the matrix. A 36 x 36 case took 3.6 seconds.

ACCURACY

Accuracy depends on the particular case being run. E. Bodewig says, "a feature of the method is that it yields smaller rounding errors than the method of Gauss-Doolittle or other methods". Several random cases were compared with the Crout inversion subroutine F1 * ML F HINT and in all cases the $AA^{-1} = I$ check was as good as or better using SYMVRT.

Author:

Supervisor:

Systems:

OPTIONS

α AND $\bar{q}\alpha$ CONSTRAINTS

The program contains an option for keeping $\alpha = 0$ or $\bar{q}\alpha$ below a specified value for a given time period. IP(11) in data block 4 is used to activate this option. The magnitude of $\bar{q}\alpha$ and the time period of the constraint are specified in data block 22.

When the zero-alpha option is used, η and χ are set to zero on forward trajectories for the time period over which the constraint is to be applied. During this time period on backward trajectories, the integrals I_{ij} are held constant. This is done because the control variables are fixed and cannot effect the terminal constraint quantities.

For the $\bar{q}\alpha$ constraint, α is set equal to $(\bar{q}\alpha)/\bar{q}$ in the region where the constraint would otherwise be violated. To accomplish this, η and χ are reduced by a common percentage. On backward trajectories, the integrals I_{ij} are held constant during the interval that $\bar{q}\alpha$ is on the constraint boundary. On forward trajectories, $\delta\eta$ and $\delta\chi$ are computed as if there were no constraint. The resulting values of η and χ are then limited as mentioned above.

Note that either eta or theta may be specified as the control variable on the initial nominal trajectory. Also, the constraint is removed at the first integration point after the time specified for removal of the constraint. Thus, if it is desired to remove the constraint at the end of a first stage which burns for 100 seconds, set the time for removal of the constraint at 99.9 seconds.

FIXED FINAL STAGE OPTION

The purpose of this option is to solve for the maximum payload capability of a configuration which has a fixed burn time in all stages. Each forward trajectory terminates as it normally would when the intended value of the stopping parameter is reached. However, the payload and thus the gross weight, is adjusted at the beginning of each trajectory so that all the propellant in the final stage should be used when the stopping condition is satisfied.

The option is called by setting IP(14)=1 in data block 4. Stage weights should be computed using the best possible estimate of payload.

At the end of the first forward guided run, the fuel remaining in the last stage is computed. This quantity is called DWFUEL. If DWFUEL is less than 1% of the final weight, the guided run is successful, assuming the other terminal constraints have been satisfied. If DWFUEL is greater than 1% of the final weight, the forward guided run is unsuccessful and a backward guidance run is made. At the end of this backward run, the change in payload required to zero DWFUEL, while meeting terminal conditions, is computed. This quantity is called DWP. DWP is computed as follows.

The changes in the control variable and adjustable parameters required to meet terminal conditions with no constraint on final weight are given by

$$\delta\alpha = W^{-1}\Lambda^T\left[\int_{t_f}^{t_i}\Lambda W^{-1}\Lambda^T dt - SY^{-1}S^T\right]^{-1}\left[d\psi - \lambda\delta x\right] \qquad (1)$$

$$\delta\tau = Y^{-1}S^T\left[\int_{t_f}^{t_i}\Lambda W^{-1}\Lambda^T dt - SY^{-1}S^T\right]^{-1}\left[d\psi - \lambda\delta x\right] \qquad (2)$$

The equations are identical to the basic control equations with the exception that the first row and/or column of all matrices are left out.

The change in final mass that will be produced by given changes in initial conditions, control variables and adjustable parameters is

$$\delta m_f = \lambda_1 \delta X + \int_{t_f}^{t_i} \Lambda_1 \delta\alpha \, dt + S_1 \delta\tau \qquad (3)$$

where λ_1 is the first row of the λ matrix

Λ_1 is the first row of the Λ matrix

S_1 is the first row of the S matrix

$\delta\alpha$ and $\delta\tau$ required to meet terminal conditions are given in Equations (1) and (2). Substitute these values into Equation (3).

$$\delta m_f = \lambda_1 \delta X + \left[\int_{t_f}^{t_i} \Lambda_1 W^{-1} \Lambda^T dt - S_1 Y^{-1} S^T \right] A \left[d\psi - \lambda \delta X \right] \qquad (4)$$

where A is the inverted matrix appearing in the brackets in Equations (1) and (2).

Equation (4) gives the changes in final mass associated with adjusting $\delta\alpha$ and $\delta\tau$ in order to meet terminal conditions in the presence of an initial perturbation δX. The only component of δX that is not zero is δm_i, the adjustment in launch mass that is to be determined.

Let $$F = \left[\int_{t_f}^{t_i} \Lambda_1 W^{-1} \Lambda^T dt - S_1 Y^{-1} S^T \right] A$$

and $G = F\lambda$

Then Equation (4) may be written as

$$\delta m_f = \left(\lambda_{m_1} - G_5 \right) \delta m_i + F \, d\psi \qquad (5)$$

where λ_{m_1} is in the fifth column of the row matrix λ_1 and G_5 is in the fifth column of the row matrix G.

The net change in final mass, assuming DWFUEL is to be zeroed is

$$\delta m_f = \delta m_i - \frac{DWFUEL}{g_0} \qquad (6)$$

Substitute this expression into Equation (5) and solve for DWP where $DWP = g_0 \delta m_i$. The result is

$$DWP = \frac{DWFUEL + g_0 F d\psi}{1 - \lambda_{m_1} + G_5}$$

This calculation is made in the MEQ subroutine at the end of every backward guidance run after the first one.

On forward optimization runs the change in launch weight is set equal to the sum of the fuel remaining on the last nominal plus the expected increase in final weight; i.e.,

$$DWP = DWFUEL + d\psi_1$$

Thus, if the expected improvement is obtained, there will be no fuel left at the time the stopping condition is reached. The quantity $d\psi_1$ is printed out as DW FINAL.

At the end of each backward guidance run, the partial derivatives relating DWFUEL and $d\psi_i \; (i=2,6)$ to DWP are printed out. These quantities are obtained from Eq. (7). The effect on payload of any errors in fuel or terminal constraints on the last guided run may be determined through use of these partial derivatives.

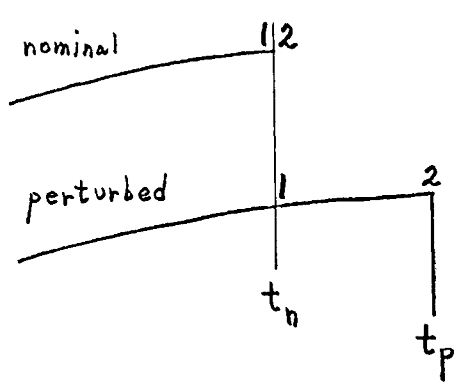
CIRCULAR PARK ORBIT OPTION

The purpose of this option is to introduce a circular park orbit at or above a specified altitude. This option is activated by specifying stage 8 as the park orbit coast stage. The stage preceding coast stage 8 must be one of the first two burns of stage 4. This burn duration is <u>not</u> to be selected as an adjustable parameter. Adjustment of coast stage 8 duration is optional. The minimum allowable altitude for the park orbit is specified in data block 23.

The fourth stage burn preceding coast stage 8 is terminated when local circular orbital velocity is reached. A constraint on inertial flight path angle equal zero is automatically introduced at the end of this burn. If the altitude of the park orbit falls below the specified minimum level on a successful guidance or optimization run, then a constraint is placed on the altitude. The permitted deviation from these two intermediate constraints are specified in data block 29, immediately following the permitted deviation in the terminal constraints. The permitted deviation in inertial flight path angle comes first. It is suggested that .1 degree and 10,000 feet be used for the deviations in path angle and altitude, respectively.

Because of the fact that the burn preceding coast stage 8 terminates when a stopping condition is reached, rather than at a fixed time, there is a discontinuity in the adjoint variables at that point. This discontinuity arises because deviations in certain trajectory variables affect the time at which the burn will end. This change in burn length, in turn, affects the terminal constraints. The method for computing the discontinuity is indicated below.

Let S, a function of the trajectory variables, be the stopping parameter. The nominal trajectory reaches S at time t_n. The perturbed



trajectory reaches S at time t_p. Point 1 occurs at t_n for both the nominal and perturbed trajectories. Point 2 represents the time at the beginning of the coast for both trajectories.

Consider the basic perturbation equation

$$d\psi = \lambda \delta \chi + \int_{t_f}^{t} \lambda \delta_\alpha \, dt + S \delta \tau \tag{1}$$

For a specified perturbed trajectory, $d\psi$ is constant regardless of the time at which (1) is evaluated. On the nominal trajectory, points 1 and 2 are coincident. The contribution of the last two terms on the right hand side of Eq. (1) to $d\psi$ is, therefore, the same at points 1 and 2. If $d\psi$ is to remain constant, $\lambda^1 \delta \chi^1$ must equal $\lambda^2 \delta \chi^2$, where the superscripts denote points 1 and 2.

$\delta \chi^2$ is found from the equation

$$\delta \chi^2 = \delta \chi^1 + \dot{\chi}^1 \delta t \tag{2}$$

where $\dot{\chi}^1$ is the rate of change of trajectory variables at 1 and

$$\delta t = t_p - t_n$$

Let δS be the change in the stopping parameter at t_n on the perturbed trajectory. Then

$$\delta t = - \frac{\delta S}{\dot{S}^1} \tag{3}$$

where $\dot{S}^1$ is the rate of change of S at point 1 on the nominal trajectory. Substitute Eq. (3) into Eq. (2)

$$\delta \chi^2 = \delta \chi^1 - \frac{\dot{\chi}^1}{\dot{S}^1} \delta S \tag{4}$$

Substitute Eq. (4) into the expression $\lambda^1 \delta \chi^1 = \lambda^2 \delta \chi^2$

$$\lambda^1 \delta \chi^1 = \lambda^2 \delta \chi^1 - \lambda^2 \frac{\dot{\chi}^1}{\dot{S}^1} \delta S \tag{5}$$

δS may be written as

$$\delta S = \frac{\partial S}{\partial X} \delta X^1 \tag{6}$$

Substitute Eq. (6) into Eq. (5) to give

$$\lambda^1 \delta X^1 = \lambda^2 \left[I - \frac{\dot{X}^1}{\dot{S}^1} \frac{\partial S}{\partial X} \right] \delta X^1 \tag{7}$$

The coefficients of δX^1 are the same on both sides of Eq. (7). Therefore,

$$\lambda^1 = \lambda^2 - \lambda^2 \frac{\dot{X}^1}{\dot{S}^1} \frac{\partial S}{\partial X} \tag{8}$$

The adjoint equations are integrated backwards in time. λ^2 is therefore known. $\frac{\partial S}{\partial X}$ and $\dot{S}^1$ are evaluated on the backward trajectory. $\dot{X}^1$ is stored on forward trajectories.

For the case at hand, the stopping condition is that

$$S = V_I - \sqrt{\frac{\mu}{r}} \tag{9}$$

Then

$$\frac{\partial S}{\partial X} = \left[\frac{\partial V_I}{\partial V} \quad \frac{\partial V_I}{\partial \gamma} \quad \left(\frac{\partial V_I}{\partial r} + \frac{V_c}{2r} \right) \quad 0 \quad 0 \quad \frac{\partial V_I}{\partial \psi} \quad \frac{\partial V_I}{\partial \lambda} \right]$$

and

$$\dot{S}^1 = \frac{\partial S}{\partial X} \dot{X}^1$$

where V_c is local circular orbital velocity.

TOTAL FLIGHT TIME CONSTRAINT

For trajectories containing at least one coast, it is possible to constrain the total flight time. IP(13) in data block 4 is used to activate this option. Total flight time is specified in data block 17. This option can be used on all six missions.

The length of the last coast in the trajectory is used to meet the total time constraint. The time change in this coast is set equal to the negative sum of the changes in earlier coasts plus the error in total time on the last forward trajectory, i.e.

$$\delta \tau_v = -\left(\sum_k \delta \tau_k + \text{time on last forward} - \text{desired total time}\right)$$

where the subscript V denotes the last coast.

Because changes in all earlier coasts are compensated for by a change in the final coast, the S vector associated with each early coast is changed. For a coast corresponding to adjustable parameter k, the term S_{ik} is replaced by $S_{ik}-S_{iv}$.

The last coast is included as an adjustable parameter in data block 6. However, the S vector associated with this coast is set to zero by the program.

The $d\psi$ vector (corrections in terminal constraints) must be adjusted to account for the change in the last coast associated with the error in total time on the last forward trajectory. If this error is denoted by δT, the quantity $S_{iv}\delta T$ is subtracted from $d\psi_i$ before the last coast. After the last coast, each $d\psi_i$ is restored to its original value.

Note that stage 5 (the integrated coast) may not be used as the last coast.

COAST RADIUS OF PERIGEE CONSTRAINT

This constraint is used to prevent the altitude during the last coast, in the input stage sequence, from falling below a specified value. The constraint is applied if, and only if, closed-form coast stage 9 is used for the last coast stage. When activated, the constraint is placed on the coast perigee altitude. The minimum allowable altitude is specified in data block 23. In addition, a permitted deviation from the minimum allowable perigee altitude must be specified in data block 29, as is done for the selected terminal constraints. This number is inserted immediately after the last number appearing in this block; i.e., if there are two terminal constraints to be met, the additional number would be placed in the third position. Also, the transition from closed-loop to open loop operation must occur before the last coast.

At the end of the last coast stage, a series of tests are made to determine whether the altitude during the coast violated the minimum allowable level. If the coast radius of perigee is above this minimum level, the altitude during coast will be satisfactory. However, if the radius of perigee is below the minimum allowable value, the perigee constraint will be activated unless the coast has all of the following properties:

1) The altitude at both ends of the coast must be above the minimum.
2) The flight path angle must be positive at the beginning of coast.
3) The coast angle must be less than 180°.

If any of these conditions is not met, and if the forward run is a guidance run that meets terminal conditions or a successful optimization run, then a radius of perigee constraint is applied to the last coast.

After the program decides that a perigee constraint is required, a backward guidance run is made. The number of terminal constraints is increased by one to make room for the additional constraint. At the end of the powered stage preceding the last coast, the initial conditions for the adjoint variables associated with the perigee constraint are introduced. Thus, the perigee constraint is identical to any other terminal constraint except that it is to be satisfied at the beginning of the last coast.

"The aeronautical and space activities of the United States shall be conducted so as to contribute . . . to the expansion of human knowledge of phenomena in the atmosphere and space. The Administration shall provide for the widest practicable and appropriate dissemination of information concerning its activities and the results thereof."

—NATIONAL AERONAUTICS AND SPACE ACT OF 1958

NASA SCIENTIFIC AND TECHNICAL PUBLICATIONS

TECHNICAL REPORTS: Scientific and technical information considered important, complete, and a lasting contribution to existing knowledge.

TECHNICAL NOTES: Information less broad in scope but nevertheless of importance as a contribution to existing knowledge.

TECHNICAL MEMORANDUMS: Information receiving limited distribution because of preliminary data, security classification, or other reasons.

CONTRACTOR REPORTS: Technical information generated in connection with a NASA contract or grant and released under NASA auspices.

TECHNICAL TRANSLATIONS: Information published in a foreign language considered to merit NASA distribution in English.

TECHNICAL REPRINTS: Information derived from NASA activities and initially published in the form of journal articles.

SPECIAL PUBLICATIONS: Information derived from or of value to NASA activities but not necessarily reporting the results of individual NASA-programmed scientific efforts. Publications include conference proceedings, monographs, data compilations, handbooks, sourcebooks, and special bibliographies.

Details on the availability of these publications may be obtained from:

SCIENTIFIC AND TECHNICAL INFORMATION DIVISION

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Washington, D.C. 20546